Statistical Machine Learning for Data Mining and Collaborative Multimedia Retrieval

HOI, Chu Hong (Steven)

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

Thesis/Assessment Committee Members

Professor Tien-Tsin Wong (Chair)

Professor Michael R. Lyu (Thesis Supervisor)

Professor Leo Jiaya Jia (Committee Member)

Professor Edward Y. Chang (External Examiner)

Abstract of thesis entitled:

Statistical Machine Learning for Data Mining and Collaborative Multimedia Retrieval

Submitted by HOI, Chu Hong (Steven)

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in September 2006

Statistical machine learning techniques have been widely applied in data mining and multimedia information retrieval. While traditional methods, such as supervised learning, unsupervised learning, and active learning, have been extensively studied separately, there are few comprehensive schemes to investigate these techniques in a unified approach. This thesis proposes a unified learning paradigm (ULP) framework that integrates several machine learning techniques including supervised learning, unsupervised learning, semi-supervised learning, active learning and metric learning in a synergistic way to maximize the effectiveness of a learning task.

Based on this unified learning framework, a novel scheme is suggested for learning *Unified Kernel Machines* (UKM). The UKM scheme combines supervised kernel machine learning, unsupervised kernel design, semi-supervised kernel learning, and active learning in an effective fashion. A key component in the UKM scheme is to learn kernels from both labeled and unlabeled data. To this purpose, a new *Spectral Kernel Learning* (SKL) algorithm is proposed, which is related to a quadratic program. Empirical results show that the UKM technique is promising for classification tasks.

Within the unified learning framework, this thesis further explores two important challenging tasks. One is *Batch Mode Active Learning* (BMAL). In contrast to traditional approaches, the BMAL method searches a batch of informative examples for labeling. To develop an effective algorithm, the BMAL task is formulated into a convex optimization problem and a novel bound optimization algorithm is proposed to efficiently solve it with global optima. Extensive evaluations on text categorization tasks show that the BMAL algorithm is superior to traditional methods.

Another issue studied in the framework is *Distance Metric Learning* (DML). Learning distance metrics is critical to many machine learning tasks, especially when contextual information is available. To learn effective metrics from pairwise contextual constraints, two novel methods, *Discriminative Component Analysis* (DCA) and Kernel DCA, are proposed to learn both linear and nonlinear distance metrics. Empirical results on data clustering validate the advantages of the algorithms.

In addition to the above methodologies, this thesis also addresses some practical issues in applying machine learning techniques to real-world applications. For example, in a time-dependent data mining application, in order to design a domain-specific kernel, marginalized kernel techniques are suggested to formulate an effective kernel aimed at web data mining tasks.

Last, the thesis investigates statistical machine learning techniques with applications to multimedia retrieval and addresses some practical issues, such as robustness to noise and scalability. To bridge semantic gap issues of multimedia retrieval, a *Collaborative Multimedia Retrieval* (CMR) scheme is proposed to exploit historical log data of users' relevance feedback for improving retrieval tasks. Two types of learning tasks in the CMR scheme are identified and two innovative algorithms are proposed to effectively solve the problems respectively.

統計機器學習在數據挖掘和協作多媒體檢索的研究

許主洪

統計機器學習近年來已經廣泛應用于數據挖掘和多媒體信息檢索。儘管傳統的學習方法論，比如監督學習、無監督學習和主動學習等，已經分別深入探討。在研究領域，迄今尚未有一個比較完善的方案可以將這些方法論有機地結合在一起。本論文提出一個統一的學習模型(ULP)來解決這個難題。該模型可以整合多種學習方法，包括監督學習、無監督學習和主動學習等，在一個有機的學習框架裏，以最有效的完成學習任務。

基於統一學習模型的思想框架，本論文提出一個新穎的方案應用于學習統一的核機器(UKM)。統一核機器結合監督核機器學習、無監督核設計、半監督核學習和主動學習在一個有機的整體。統一核機器的一個關鍵的部件是同時從標定數據和無標定數據中學習一個有效的核函數或核矩陣。本論文提出一種新的算法名為核譜學習算法(SKL)。它等價于一個二次規劃的優化問題，可以很有效地實現。從數据分類的實驗結果可以看出，本論文提出的方案和算法比傳統方法更加有效。

在統一學習的框架，本論文特別針對兩個重要難題作深入探討。一個是主動學習問題。傳統的方法通常在每個學習過程只考慮選擇一個樣本作標定，我們提出批量主動學習的方法(BMAL)，可以搜索一批最有信息量的樣本讓用戶標定，使得大規模的分類任務更加有效地完成。爲了設計一個有效的算法，本論文把批量主動學習形式化為凸優化問題，然後提出一個逼近算法有效找出最優解。通過在文本分類實驗的詳細評估，我們發現批量主動學習算法在大多數情況下要比傳統算法更加有效。

另外一個重要的難題是距離尺度學習的問題(DML)。學習有效距離尺度是很多機器學習算法的本質問題。本論文探討如何從上下文的兩兩約束數據裏學習出有效的距離尺度。本論文提出判別分量分析(DCA)和它的核擴展算法，也就是核判別分量分析(KDCA)，來解決綫性和非綫性的尺度學習。相比其它複雜的方法，我們提出的算法相當簡單，而且從數據聚類的實驗上觀察，效果相當理想。

除了以上的方法論研究，本論文也研究如何應用統計學習方法論解決現實世界的一些問題。一個應用是使用核方法解決網絡搜索引擎的查詢日誌挖掘問題。我們提出用邊際核技術來設計尺度測量的算法，使得更有效地從查詢日誌裏找出與時間相關的查詢模式。另外一個重用的應用是協作多媒體檢索(CMR)問題。跟傳統多媒體檢索方法不一樣，我們提出利用用戶的相關反饋日誌來協作當前的檢索任務。從長期的學習考慮，這是一種有效的方案來解決多媒體檢索的語意差距難題。爲了更魯棒地處理協作多媒體檢索，提出提出幾個可靠和可擴展的機器學習算法，有效地解決多媒體檢索的一些重用難題。

Acknowledgement

I would like to thank my supervisor, Prof. Michael R. Lyu, for his persistent guidance during my graduate study. Prof. Lyu has taught me much not only on research, but also about presentation, teaching, and English writing skills, and so on. All his efforts towards my education have significantly improved many aspects of my development. Without his supervision and encouragement, I would never have achieved fruitful outputs in this thesis.

I also appreciate my research collaborators who have provided help and guidance throughout the research projects in this thesis. Prof. Edward Y. Chang at UCSB, my thesis committee member, has offered nice instructions for my research. I am glad that we have collaborated in some nice work. Prof. Rong Jin at MSU has given much help to my research. We have collaborated on several pieces of great work. I also want to thank my other collaborators, Luo Si at CMU, Qiankun Zhao and Wei-Ying Ma at Microsoft, for our nice collaborations.

I also thank other thesis committee members, Prof. T.T. Wong and Prof. Leo Jia, for their nice comments and great tutoring experiences with them. I would also like to thank my colleagues in our group including Prof. Irwin King, Jianke Zhu, Haixuan Yang, and others, for nice discussions. I also thank other colleagues and friends at CUHK for the joyful time I have had with them in the past four years.

Finally, I am deeply grateful to my parents for their unlimited love and tolerance in supporting my Ph.D study.

This work is dedicated to my beloved parents and family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statistical Machine Learning

1.1.1 Overview

Statistical machine learning was introduced in the late 1960's. Until the 1990's it was still almost a purely theoretical analysis of the problem of function estimation from a given collection of data. However, in the middle of the 1990's some breakthroughs were achieved. At that time, a new type of algorithms, e.g. support vector machines, achieved quite exciting successes in a variety of applications [141]. These successes of the emerging new type of algorithms based on statistical learning theory showed that statistical machine learning not only can be a tool for theoretical analysis but also can be used to develop effective algorithms for solving real-world problems. In the past decade, there has been a surge of interest in studying statistical machine learning techniques for a variety of real-world applications, particularly in data mining and information retrieval.

In general, statistical machine learning studies a variety of different classes of problems. In terms of different settings and ways of studying the methodology, they can typically be categorized as *unsupervised learning*, *supervised learning*, *semi-supervised learning*, and *active learning*, as well as others. Each of them has been separately studied in the past few years. Let us briefly introduce each of them as follows.

1.1.2 Unsupervised Learning

Unsupervised learning considers the problem of learning from a collection of data instances without training labels. It intends to discover the cluster patterns among the given collection of data. One of the most popular areas of study in unsupervised learning is data clustering techniques, which have been widely used for data mining applications [64]. Despite the fact that this problem has been studied for many years and many algorithms have been proposed, many challenging problems in unsupervised learning are still actively being studied in current research communities.

1.1.3 Supervised Learning

Supervised learning considers the problems of estimating certain functions from examples with label information. Each input data instance is associated with some corresponding training label, which is assumed to be the response from a supervisor. A broad family of statistical learning theories has been studied to achieve the risk minimization and generalization maximization in the learning tasks. These theories have guided the creation of many new types of supervised learning algorithms for applications. Among them, supervised kernel-machine learning is the state-of-the-art methodology in real-world applications. Many algorithms, such as Support Vector Machines (SVM) and Kernel Logistic Regression (KLR), have shown excellent performance in a range of applications, especially in classification and regression tasks.

1.1.4 Active Learning

For a supervised learning task, the training examples can be expensive to obtain. In order to look for the most informative example for labeling, active learning has been introduced as an important technique to minimize the human efforts in finding the most informative examples for manual labeling. Active learning has already been employed as an important tool for reducing the human effort in a number of classification tasks [29].

1.1.5 Semi-Supervised Learning

One issue for supervised learning is the problem of learning when there are insufficient training examples. To take advantage of both labeled and unlabeled data, semi-supervised learning has recently been proposed to address the challenge of learning from small number of training samples. It has been demonstrated to be a promising approach, affording improved performance compared with traditional supervised learning approaches when only limited training examples are offered [26].

1.1.6 Distance Metric Learning

Statistical machine-learning techniques, such as K-Means and K-Nearest Neighbor, usually define some distance metrics or kernel functions to measure the similarity of data instances. For example, Euclidean distance is often used as a distance measure in many applications. Typically, selection of a good quality distance metric can enhance the performance of the learning algorithm significantly. Therefore, determining how to learn an appropriate distance metric for various learning algorithms has been an open issue in recent research [9]. Distance metric learning techniques can be applied for a wide range of applications in data mining and multimedia retrieval, such as clustering, classification, and content-based image and video retrieval [53].

1.2 Unified Learning Paradigm

1.2.1 Motivation

Human beings learn by being taught (supervised learning), by self-study (unsupervised learning), by asking questions (active learning), and by being examined for the ability to generalize (metric learning or reinforcement learning), among many ways of acquiring knowledge. An integrated process of supervised learning, unsupervised learning, active learning, metric learning and reinforcement learning provides a foundation for acquiring the known and discovering the unknown.

It is natural to extend the human learning process to statistical machine learning tasks. To this purpose, this thesis proposes a framework of unified learning paradigm (ULP), which combines several machine-learning techniques in a synergistic way to maximize the effectiveness of a learning task. Three characteristics distinguish ULP from a traditional hybrid approach. First, ULP aims to minimize the human effort required for the collection of quality labeled data. Second, ULP uses the cluster information of unlabeled data, together with semi-supervised learning, to ensure sufficiency of both labeled and unlabeled data, thus guaranteeing the generalization ability of the learned results. Third, ULP uses active learning and metric learning (or reinforcement learning and some other techniques) to access and speedup the convergence of the learning process.

1.2.2 The Unified Learning Framework

Figure 1.1 illustrates the architecture of the Unified Learning Paradigm. Basically, the ULP scheme comprises several main components including a *kernel initialization* module, an *unsupervised learning* module, a *semi-supervised kernel learning* module, an *active learning* module, and a *distance metric learning* module. The unsupervised learning module learns the clustering information among the unlabeled data. If contextual information is also available, the unsupervised learning module can use the metric learned from the context via distance metric learning techniques. After the unsupervised learning module has finished, the clustering results are transmitted to the semi-supervised kernel learning module. This collects the information from both labeled data and unlabeled data, as well as contextual data, to learn an effective kernel function or matrix. Once the kernel is achieved, a supervised kernel machine learning technique is adopted to train a unified kernel machine based on the established kernels. Given that the trained kernel machine may not be good enough for the final application, the active learning module is used to choose a set of the most informative unlabeled examples for labeling by users. Finally, convergence will be tested on the unified kernel machines. If the resulting kernel machine passes the test, then the ULP algorithm ends; otherwise it repeats the above procedures for another iteration.

1.2.3 Open Issues

The above architecture shows a general ULP framework in terms of kernel machine formulation. Several challenging issues that will be explored in this thesis include:

(1) **Semi-Supervised Kernel Learning.**
The goal of the semi-supervised kernel learning module is to learn effective kernel metrics from a collection of data. For a data classification task, given labeled and unlabeled data, this can be regarded as a problem of learning semi-supervised kernels from the data. This is an important issue addressed in the present work.

(2) **Batch Mode Active Learning.**
Active learning is critical to speeding up a learning task effectively.

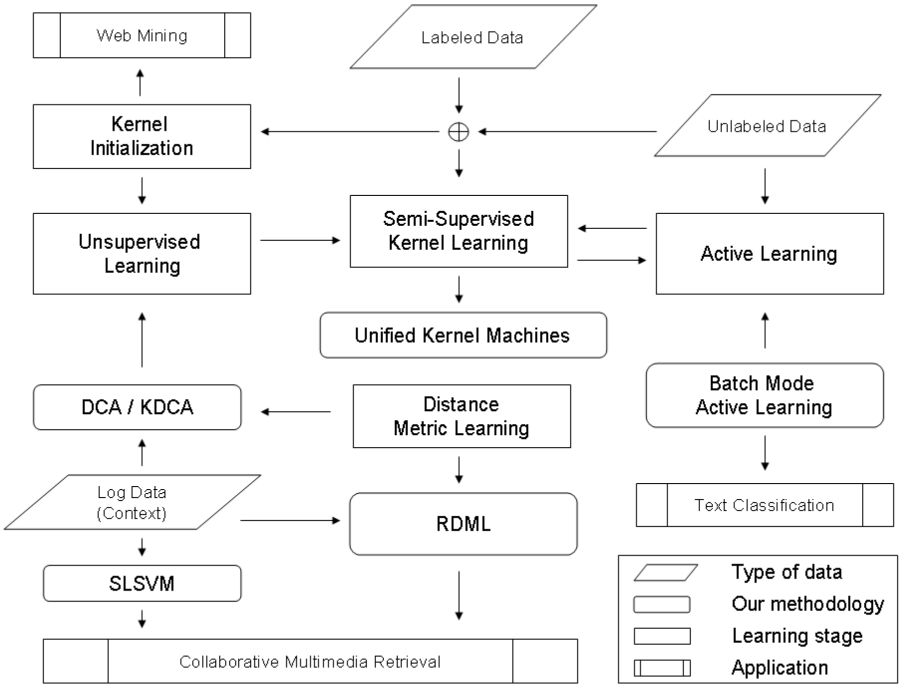


Figure 1.1: The unified learning paradigm framework

However, it is not yet clear how to develop an effective active learning algorithm. This thesis proposes a Batch Mode Active Learning (BMAL) method, which searches a batch of the most informative examples for labeling. This may be more effective than traditional methods.

(3) **Distance Metric Learning.**
When users' contextual information is available, the distance metric learning module has to learn effective distance metrics from contexts. The metric can be either a linear Mahalanobis metric or a kernel metric. Learning a quality metric is essential to accomplishing the learning task.

These three components play the key roles in the ULP framework. This thesis mainly focus on these issues. There are some other open issues that need to be explored in the future, such as theories of convergence and generalization performances.

1.3 Applications

In addition to the studies of statistical machine learning methodology, this thesis also investigates these techniques and algorithms with applications to real-world problems. Two main applications are studied. One is data mining and web search application. The other is collaborative multimedia retrieval. Let us briefly introduce the main applications and related problems which will be explored in this thesis.

1.3.1 Data Mining and Web Applications

Two classical data mining problems considered here are *data clustering* and *classification* tasks. This thesis investigates these problems with extensions to web based applications including text categorization and web query log mining.

Text Categorization. The goal of text categorization is to automatically assign text documents to predefined categories. With the rapid growth of Web pages on the World Wide Web (WWW), text categorization has become more and more important, both in the world of research and for user applications. Usually, text categorization is regarded as a supervised learning problem. In order to build a reliable model for text categorization, we need to first of all manually label a number of documents using the predefined categories. Then, a statistical machine learning algorithm is engaged to learn a text classification model from the labeled documents. One important challenge for large-scale text categorization is how to reduce the number of labeled documents that are required for building reliable text classification models. This is particularly important for text categorization of WWW documents given the huge number of documents available on the Web. To reduce the number of labeled documents, a novel batch mode active learning scheme is proposed, which is able to select a *batch* of the most informative unlabeled examples in each learning round.

Web Query Log Mining. One real dilemma for most of the existing Web search engines is that users expect accurate search results while they only provide queries of limited length, which is usually less than two words on average, according to the study in [147]. Recently, a lot of work has been done in the Web search community to expand the query terms with similar keywords for refining the search results [10, 32, 147, 155, 156]. The basic idea is to use the click-through data, which records the interactions between users and a search engine, as the feedback to learn the similarity between query keywords. In this thesis, a time-dependent framework is suggested to measure the semantic similarity between Web search queries by mining the click-through data. More specifically, a novel *time-dependent query term semantic similarity model* is proposed and formulated by a*Marginalized Kernel* technique that can exploit the click-through data more effectively than traditional approaches.

1.3.2 Collaborative Multimedia Retrieval

The second main application considered in this thesis is multimedia information retrieval. I restrict the attention to content-based image retrieval (CBIR). CBIR has been an active research topic in the last decade [43, 88, 124]. Although substantial research has been conducted, CBIR is still an open research topic, mainly due to the difficulty in bridging the gap between low-level feature representation and high-level semantic interpretation. Several approaches have been proposed to reduce the semantic gap and to improve the retrieval accuracy of CBIR systems.

One promising approach is to use online relevance feedback [5, 30, 48, 49, 50, 52, 55, 57, 56, 61, 102, 109, 110, 136]. This method first solicits users' relevance judgments on the initial retrieval results for a given query image. It then refines the representation of the initial query according to the acquired user judgments, and re-runs the CBIR algorithm again with the refined representation. Given the difficulty in learning users' information needs from their relevance feedback, multiple rounds of relevance feedback are usually required before satisfactory results are achieved, which can significantly limit the application of this approach to real-world problems.

An alternative approach to bypass the semantic gap is to index image databases with text descriptions and allow users to pose textual queries against image databases. To avoid the excessive amount of labor involved in manual annotation, automatic image annotation techniques have been developed [17, 36, 66, 81]. However, text descriptions generated by automatic annotation techniques are often inaccurate and limited to a small vocabulary, and are therefore insufficient to accommodate the diverse information needs of users.

Another important way to bridge the semantic gap is to exploit users' relevance feedback logs, an approach which has received only a little attention in the community. From a long-term learning perspective, users' feedback log data is an important resource to aid the retrieval task in CBIR. To this end, a framework of "Collaborative Multimedia Retrieval (CMR)" is proposed, which exploits users' relevance feedback log data for improving retrieval tasks. In order to develop effective solutions for different retrieval stages, a two-stage learning scheme is suggested for CMR. One is "Online Collaborative Multimedia Retrieval", which learns online relevance feedback with users' feedback logs based on a unified log-based relevance feedback scheme. The other is "Offline Collaborative Multimedia Retrieval", which learns a distance metric offline from users' relevance feedback logs. These two schemes can collaborate in a unified solution in order to accomplish the retrieval tasks effectively.

1.4 Contributions

This thesis aims to develop a unified solution that can combine several statistical machine learning techniques in an effective learning way. To this purpose, a novel framework of unified learning paradigm (ULP) is presented, which integrates several learning techniques including supervised learning, unsupervised learning and active learning in a synergistic way to maximize the effectiveness with which a learning task is carried out. Based on this global framework, some challenging problems are addressed and novel algorithms are proposed to solve them effectively. The main contributions of this thesis can be further described as follows:

(1) *Learning Unified Kernel Machines.*

A novel classification scheme of learning unified kernel machines is proposed, which combines supervised kernel-machine learning, unsupervised kernel design and active learning in a unified solution. In this scheme, a new *Spectral Kernel Leaning* (SKL) algorithm is developed, which learns more effective semi-supervised kernels from labeled and unlabeled data than traditional semi-supervised kernel learning methods.

(2) *Batch Mode Active Learning.*

A novel framework of Batch Mode Active Learning (BMAL) is proposed and formulated into a convex optimization problem. To solve the problem efficiently, an efficient BMAL algorithm is developed that can solve large-scale problems effectively. Extensive evaluations are conducted on text categorization tasks and promising results are found.

(3) *Distance Metric Learning.*

To learn effective distance metrics from context, two new algorithms, *Discriminative Component Analysis* (DCA) and *Kernel Discriminative Component Analysis* (KDCA), are proposed to learn both linear and nonlinear distance metrics from pairwise contextual constraints. The proposed methods enjoy the merits of simplicity and state-of-the-art performance in data clustering applications.

(4) *Marginalized Kernels for Web Query Mining.*

To tackle the challenge of mining web query logs to improve web searches, a novel *time-dependent framework* is proposed for mining semantic related queries from web query log data. To develop an effective algorithm, *Marginalized kernel* techniques are suggested to design kernels that can measure the similarities between queries effectively. The suggested method has been shown effective from extensive evaluations on query log data collected from a real-world search engine.

(5) *Online Collaborative Multimedia Retrieval.*

To attack the challenging semantic gap problem, an "*Online Collaborative Multimedia Retrieval*" scheme is studied, which learns online relevance feedback with users' feedback logs based on a novel *Log-based Relevance Feedback* (LRF) solution. To develop an effective LRF algorithm, a modified SVM technique, called *Soft Label Support Vector Machine* (SLSVM), is proposed, which can solve the LRF task more effectively and robustly.

(6) *Offline Collaborative Multimedia Retrieval.*

In a long-term consideration, to bridge the semantic gap and reduce the online learning cost, an "*Offline Collaborative Multi-*

media Retrieval is investigated, which learns a reliable distance metric offline by using a novel Regularized Distance Metric Learning (RDML) algorithm. Compared with traditional methods, the RDML algorithm is able to learn more robust metrics, particularly in the presence of noisy log data.

1.5 Scope and Organization

This thesis reviews some main methodology in statistical machine learning, and presents a framework of unified learning paradigm that integrates several machine learning techniques in a unified solution. Based on the framework, several important issues including distance metric learning and batch mode active learning, are extensively explored. This thesis also extends some novel statistical machine learning techniques to address some real-world problems in web data mining and multimedia retrieval applications and demonstrate promising results. The rest of this thesis is organized as follows:

- Chapter 2

  This chapter reviews some background knowledge and work related to the main methodology and problems that will be discussed in this thesis.

- Chapter 3

  I present a scheme for learning unified kernel machines (UKM) for classification tasks based on the ULP solution. A new spectral kernel learning algorithm is proposed to learn semi-supervised kernels from both labeled and unlabeled data. Then the UKM scheme is formulated and applied to learn a paradigm of Unified Kernel Logistic Regression for classification tasks. Empirical results on benchmark datasets will be discussed.

- Chapter 4

  This chapter investigates the problem of active learning to search a batch of informative examples for labeling. To do active learning effectively for large-scale applications, a scheme of batch mode active learning (BMAL) is proposed and formulated into a convex optimization problem that can be efficiently solved. Extensive evaluations on text categorization will be discussed.

- Chapter 5

  This chapter studies the problem of distance metric learning. To learn effective metrics from pairwise contextual constraints, two novel algorithms, Discriminative Component Analysis (DCA) and Kernel DCA, are proposed to learn both linear and nonlinear distance metrics. Empirical evaluations on data clustering will be discussed.

- Chapter 6

  This chapter applies kernel techniques to solve a real-world web data mining problem. A new time-dependent framework is suggested for mining semantic related queries from users' query log data, which is formulated using marginalized kernel techniques. Extensive evaluations on real-world web query logs will be discussed.

- Chapter 7

  This chapter studies the problem of online collaborative multimedia retrieval by applying supervised kernel machine learning techniques. The problem is tackled by means of a log-based relevance feedback scheme formulated using a novel soft label support vector machine algorithm, which is more robust for noisy data. Extensive evaluations on real-world log data will be studied.

- Chapter 8

  This chapter studies the problem of offline collaborative multimedia retrieval by applying distance metric learning techniques. The problem is solved by a Regularized Distance Metric Learning (RDML) algorithm, which is formulated as a semi-definite programming problem that can be solved with global optima. Compared with traditional methods, the RDML algorithm is more reliable at learning a robust metric. Extensive evaluations on real-world log data will be studied.

- Chapter 9

  The last chapter summarizes the achievements of this thesis and addresses some directions to be explored in future work.

Each chapter of the thesis is intended to be self-contained. Thus, in some chapters, some definitions, formulas, or illustrative figures that have already appeared in previous chapters, may be briefly reiterated.

□ **End of chapter.**

Chapter 2

Background Review

2.1 Supervised Learning

Supervised learning considers the problem of learning the function that best estimates the responses of a supervisor given a collection of training examples. There is a rich basis of statistical learning theory for the supervised learning setting. Let us first take a short review of statistical learning theory in terms of regularization theory.

2.1.1 Overview of Statistical Learning Theory

In a general setting of supervised learning, assume that we are given a training set of l independent and identically distributed observations

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)$$

where $\mathbf{x}_i$ are vectors produced by a generator and y_i are the associated responses by a supervisor. A learning machine estimates a set of approximated functions f to approach the supervisor's responses. It is an ill-posed problem to approximate a function from sparse data. This is typically solved by the regularization theory [38]. The classical regularization theory formulates the learning problem as a variational problem of finding the function f which tends to minimize the following functional:

$$f = \arg \min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} L(\mathbf{x}_i, y_i, f) + \lambda \|f\|_K^2, \tag{2.1}$$

where $L(\cdot, \cdot, \cdot)$ is a loss function, $\|f\|_K^2$ is a norm in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ defined over the positive definite function K, λ is the regularization parameter, and the whole penalty norm $\lambda\|f\|_K^2$ imposes smoothness conditions on the solution space.

Many supervised kernel-machine techniques can be formulated into the above regularization learning framework. Here, let us show three different choices of loss functions which correspond to three state-of-the-art algorithms:

- Regularized Least Squares Networks (RLS):

$$V(\mathbf{x}_i, y_i, f) = (y_i - f(\mathbf{x}_i))^2 \ , \tag{2.2}$$

- Support Vector Machine Regression (SVMR):

$$V(\mathbf{x}_i, y_i, f) = (y_i - f(\mathbf{x}_i))_\epsilon \ , \tag{2.3}$$

- Support Vector Machine Classification (SVMC):

$$V(\mathbf{x}_i, y_i, f) = (1 - y_i f(\mathbf{x}_i))_+ \ , \tag{2.4}$$

where $(\cdot)_\epsilon$ is the Vapnik's epsilon-insensitive norm [141], $(\cdot)_+$ is the hinge loss in which $(a)_+ = a$ if a is positive and zero otherwise, and y_i is a real number for both RLS and SVMR, and takes +1 or -1 for SVMC. Two popular algorithms, Support Vector Machines and Kernel Logistic Regressions, will be further discussed in the following sections.

2.1.2 Support Vector Machines

Support Vector Machines (SVMs) enjoy solid theoretical foundations and have demonstrated outstanding performance in many empirical applications [21]. In theory, SVM can be interpreted from the statistical regularization learning framework [38]. More specifically, SVM can be formulated as a similar regularized learning problem:

$$f = \arg \min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} (1 - y_i f(\mathbf{x}_i))_+ + \lambda \|f\|_K^2, \tag{2.5}$$

where $(\cdot)_+$ is the hinge loss in which $(a)_+ = a$ if a is positive and zero otherwise, and y_i is the class label.

However, practical SVM users may be more familiar with another formula:

$$\begin{aligned} \min_{\mathbf{w},\xi,b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - b) \geq 1 - \xi_i \ , \\ & \xi_i \geq 0 \ , i = 1, 2, \cdots, l \ , \end{aligned} \tag{2.6}$$

where C is a penalty parameter of the error term ξ_i, which is equivalent to the regularization parameter $\frac{1}{2\lambda l}$ where λ is the parameter in the above regularization framework, $\Phi(\cdot)$ is a kernel mapping function, and the labels y_i are either $+1$ or -1 for a regular binary classification problem.

The solution to the above convex optimization problem can be found by introducing the Lagrange functional technique [141, 19]. It then can be formulated into a dual form as a QP problem as follows:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{l} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \ , i = 1, 2, \ldots, l \ . \end{aligned} \tag{2.7}$$

This is a typical QP problem that can be solved effectively by a standard QP technique or by some other available method, such as Sequential Minimal Optimization (SMO) techniques [101].

2.1.3 Kernel Logistic Regressions

Similarly, Kernel Logistic Regression (KLR) can also be formulated into the regularization learning framework:

$$\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} \ln(1 + e^{-y_i f(\mathbf{x}_i)}) + \frac{\lambda}{2} ||f||^2_{\mathcal{H}_K} \ , \tag{2.8}$$

where λ is a regularization parameter. To solve the above optimization, let us first definite the following notations:

$$\begin{aligned}
p_i &= \frac{1}{1 + e^{-y_i f(\mathbf{x}_i)}}, \quad i = 1, \dots, l & (2.9)\\
\boldsymbol{\alpha} &= (\alpha_1, \dots, \alpha_l)^\top & (2.10)\\
\boldsymbol{p} &= (p_1, \dots, p_l)^\top & (2.11)\\
\boldsymbol{y} &= (y_1, \dots, y_l)^\top & (2.12)\\
\mathbf{K}_1 &= (K(\mathbf{x}_i, \mathbf{x}'_i))^l_{i,i'=1} & (2.13)\\
\mathbf{K}_2 &= \mathbf{K}_2 & (2.14)\\
\mathbf{W} &= diag(p_1(1-p_1), \dots, p_l(1-p_l)). & (2.15)
\end{aligned}$$

Using the above notations and the representer theorem [141], the objective function can be rewritten as follows:

$$\frac{1}{l} \mathbf{1}^\top \ln(1 + e^{-\mathbf{y} \cdot (\mathbf{K}_1 \boldsymbol{\alpha})}) + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \mathbf{K}_2 \boldsymbol{\alpha}, \tag{2.16}$$

where "·" denotes an element-wise multiplication. To find the solution $\boldsymbol{\alpha}$, one can use a Newton-Raphson method to solve it iteratively by the following steps:

$$\boldsymbol{\alpha}^{(k)} = (\frac{1}{l} \mathbf{K}_1^\top \mathbf{W} \mathbf{K}_1 + \lambda \mathbf{K}_2)^{-1} \mathbf{K}_1^\top \mathbf{W} \mathbf{z}, \tag{2.17}$$

where $\boldsymbol{\alpha}^{(k)}$ is the value of $\boldsymbol{\alpha}$ in the k-th step, and $\mathbf{z}$ is computed as follows:

$$\mathbf{z} = \frac{1}{l} (\mathbf{K}_1 \boldsymbol{\alpha}^{(k-1)} + \mathbf{W}^{-1} (\mathbf{y} \cdot \mathbf{p})) \ . \tag{2.18}$$

2.2 Unsupervised Learning

In many real-world applications, it may be expensive to assign labels to data. In these situations, unsupervised learning techniques are often used to discover unknown knowledge from a large amount of unlabeled data. A well-known methodology among various unsupervised learning techniques is data clustering. Let us briefly review several representative clustering algorithms.

2.2.1 K-Means Clustering

In general, in an unsupervised learning task, assume that we are given a collection of data examples $\{\mathbf{x}_i\}_{i=1}^n$. The goal of clustering is to divide the data examples into k disjoint groups such that examples in a same group share the similar characteristics.

The K-Means clustering algorithm is one of the most popular clustering techniques. The main idea of K-Means is to divide data examples so that the within-group scatter is minimized. Typically, it proceeds via the following steps:

(1) Initialize a random partition: $\{\mathcal{C}_i\}_{i=1}^k$.

(2) Update assignments until convergence:
For each $\mathbf{x}_j$, assign $\mathbf{x}_j \to \mathcal{C}_p$
where $p = \arg\min_i \|\mathbf{x}_j - \mu_i\|_M$ and $\mu_i = \frac{1}{|\mathcal{C}_i|}\sum_{\mathbf{x}\in\mathcal{C}_i}\mathbf{x}$

where M is a distance metric among a family of Mahalanobis Metrics. The Euclidean distance metric is typically used by default for K-Means clustering.

2.2.2 Kernel K-Means Clustering

Regular K-Means using a linear metric cannot separate data with complex nonlinear relationships, such as non-convex shapes. Kernel K-Means clustering using kernel trick maps the original data to a feature space by a nonlinear transformation $\phi : \mathbf{x} \rightarrow f$, and then runs K-Means in the feature space. Typically, Kernel K-Means clustering proceeds via the following steps:

(1) Initialize a random partition: $\{\mathcal{C}_i\}_{i=1}^k$.

(2) Update assignments until convergence:
For each $\mathbf{x}_j$, assign $\mathbf{x}_j \rightarrow \mathcal{C}_p$
where $p = \arg\max_i \left(2|\mathcal{C}_i| \sum_{\mathbf{x}\in\mathcal{C}_i} K(\mathbf{x}_j, \mathbf{x}) - \sum_{\mathbf{x},\mathbf{x}'\in\mathcal{C}_i} K(\mathbf{x}, \mathbf{x}')\right)$

where K is a predefined kernel, such that $K(\mathbf{x}, \mathbf{x}') = \langle\phi(\mathbf{x}), \phi(\mathbf{x}')\rangle$.

2.3 Semi-Supervised Learning

Semi-supervised learning considers the problem of learning from both a set of labeled data pairs $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$ and a set of unlabeled data $\{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_n\}$, in which the number of unlabeled examples n-l is typically much larger than the labeled ones l. In recent research studies, many methods have been proposed for solving semi-supervised learning problems, such as EM with generative mixture models [100], co-training [18], self-training [107], Transductive Support Vector Machines [69], and graph-based methods [12, 14, 127, 173, 172], among others. A comprehensive survey can be found in [170].

2.4 Active Learning

Active learning, or so-called pool-based active learning, has been extensively studied in machine learning for many years and has already been employed for text categorization [84, 85, 93, 108]. Most active learning algorithms are conducted in an iterative fashion. In each iteration, the example with the highest classification uncertainty is chosen for labeling manually. Then, the classification model is retrained with the additional labeled example. The step of training a classification model and the step of soliciting a labeled example are iterated alternately until most of the examples can be classified with reasonably high confidence. One of the key issues in active learning is how to measure the classification uncertainty of unlabeled examples. In [39, 40, 44, 76, 93, 119], a number of distinct classification models are first generated. Then, the classification uncertainty of a test example is measured by the amount of disagreement among the ensemble of classification models in predicting the labels for the test example. Another group of approaches measures the classification uncertainty of a test example according to how far the example is away from the classification boundary (i.e., the classification margin) [22, 116, 137]. One of the most well-known approaches within this group is *support vector machine active learning*, developed by Tong and Koller [137]. Due to its popularity and success in previous studies, it is used as the baseline approach in our study.

2.5 Distance Metric Learning

The problems for learning distance metrics and data transformation have become more and more popular in recent research due to their broad application. One kind of approach is to use the class labels of data instances to learn distance metrics in supervised classification settings. Let us briefly introduce several traditional methods. Hastie et al. [47] and Jaakkola et al. [63] used the labeled data instances to learn distance metrics to address classification tasks. Tishby et al. [135] considered the joint distribution of two random variables X and Y to be known, and then learned a compact representation of X that enjoys high relevance to Y. Most recently, Goldberger et al. [42] proposed the Neighborhood Component Analysis approach to learn a distance measure for kNN classification by directly maximizing a stochastic variant of the leave-one-out kNN score on the training set. Zhou et al. proposed a kernel partial alignment scheme to learn kernel metrics for interactive image retrieval [168]. Most of these studies need to explicitly use the class labels as the side-information for learning the representations and distance metrics.

Recently, some work has addressed the problems of learning with contextual information in terms of pairwise constraints. Wagstaff et al. [145] suggested the K-means clustering algorithms by introducing the pairwise relations. Xing et al. [153] studied the problem of finding an optimal Mahalanobis metric from contextual constraints with a constrained K-means algorithm. Bar-Hillel et al. [9] proposed a much simpler approach called Relevant Component Analysis (RCA), which enjoys comparable performance with Xing's method. Other related methods studied recently can also be found in [77, 150]. Due to the popularity of RCA and Xing's methods, let us review these two important techniques as follows.

The problem of distance metric learning is to find the optimal Mahalanobis metric that is used to measure the distance between two data instances as $d_M(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top M (\mathbf{x}_i - \mathbf{x}_j)}$, where M must be positive semi-definite to satisfy the properties of a metric, i.e., non-negativity and triangle inequality. The matrix M can be decomposed as $M = A^\top A$, where A is a transformation matrix. The goal of RCA learning is to find an optimal Mahalanobis matrix M and the optimal data transformation matrix A using the contextual information.

Given the pairwise contextual constraints $\mathcal{S}$ and $\mathcal{D}$, Xing et al. [153] formulated the problem of distance metric learning into the following convex programming problem:

$$\begin{aligned} \min_{\mathbf{M}} & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{M}}^2 \\ \text{s. t.} & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{M}}^2 \geq 1 \\ & \mathbf{M} \succeq 0 \end{aligned} \tag{2.19}$$

In the equations above, the optimal metric $\mathbf{M}$ is found by minimizing the sum of the squared distances between pairs of similar data examples $\mathcal{S}$, and meanwhile satisfying the constraint that the sum of the squared distances between dissimilar data examples $\mathcal{D}$ is larger than 1. In other words, this algorithm tries to minimize the distance between similar data and maximize the distance between dissimilar data at the same time.

RCA uses a much simpler approach for distance metric learning. The basic idea of RCA learning is to identify and scale down global unwanted variability within the data. RCA changes the feature space used for data representation via a global linear transformation in which relevant dimensions are assigned with large weights [9]. The relevant dimensions are estimated by chunklets [9], each of which is defined as a group of data instances linked together with positive constraints. More specifically, given a data set $X = \{\mathbf{x}_i\}_{i=1}^N$ and n chunklets $C_j = \{\mathbf{x}_{ji}\}_{i=1}^{n_j}$, RCA computes the following matrix:

$$\hat{C} = \frac{1}{N}\sum_{j=1}^{n}\sum_{i=1}^{n_j}(\mathbf{x}_{ji} - \mathbf{m}_j)(\mathbf{x}_{ji} - \mathbf{m}_j)^\top \tag{2.20}$$

where $\mathbf{m}_j$ denotes the mean of the j-th chunklet, $\mathbf{x}_{ji}$ denotes the i-th data instance in the j-th chunklet and N is the number of data instances. The optimal linear transformation by RCA is then computed as $A = \hat{C}^{-\frac{1}{2}}$ and the Mahalanobis matrix is equal to the inverse of the matrix C, i.e., $M = \hat{C}^{-1}$. RCA enjoys the merits of simple implementation and good computational efficiency.

2.6 Web Data Mining

2.6.1 Text Categorization

The goal of text categorization is to automatically assign text documents to the predefined categories. With the rapid growth of Web pages on the World Wide Web (WWW), text categorization has become more and more important, both the world of research and in practical applications. Usually, text categorization is regarded as a supervised learning problem. In order to build a reliable model for text categorization, we need first of all to manually label a number of documents using the predefined categories. Then, a statistical machine learning algorithm is engaged to learn a text classification model from the labeled documents. One important challenge for large-scale text categorization is how to reduce the number of labeled documents that are required for building reliable text classification models. This is particularly important for text categorization of WWW documents, given the huge number of documents available on the Web.

Text categorization is a long-term research topic which has been actively studied in the communities of Web data mining, information retrieval and statistical learning [79, 158]. Essentially, the text categorization techniques have been the key toward automated categorization of large-scale Web pages and Web sites [87, 122], which is further applied to improve Web searching engines in finding relevant documents and to help users browsing Web pages or Web sites.

In the past decade, a number of statistical learning techniques have been applied to text categorization [157], including the K Nearest Neighbor approaches [92], decision trees [4], Bayesian classifiers [139], inductive rule learning [28], neural networks [112], and support vector machines (SVM) [67]. Empirical studies in recent years [67] have shown that SVM is the state-of-the-art technique among all the methods mentioned above.

Recently, logistic regression, a traditional statistical tool, has attracted considerable attention for text categorization and high-dimension data mining [74]. Several recent studies have shown that the logistic regression model can achieve comparable classification accuracy to SVMs in text categorization. Compared to SVMs, the logistic regression model has the advantage in that it is usually more efficient in model training, especially when the number of training documents is large [75, 160]. This motivates us to choose logistic regression as the basis classifier for large-scale text categorization.

2.6.2 Web Query Log Mining

In this part, I review some related work in web query log mining, mainly with respect to the following two aspects: query expansion with click-through data, and temporal analysis of click-through data.

Query expansion with click-through data is motivated by the well-known relevance feedback techniques, which modify the queries based on users' relevance judgments of the retrieved documents [6, 34, 71]. Typically, expansion terms are extracted based on the frequencies or co-occurrences of the terms from the relevant documents. However, it is difficult to obtain sufficient feedback, since users are usually reluctant to provide such feedback information. Even though the pseudo-relevance feedback approach can partially alleviate the lack of feedbacks, it still suffers from the failure of the assumption that a frequent term from the top-ranked relevant documents will tend to co-occur with all query terms, which may not always hold [8, 155].

The click-through data has been studied for query expansion in the past [10, 32, 147, 156]. The existing work can be categorized into two groups. The first group examines the approach of expanding queries with similar queries based on the assumption that similarity between queries may be deduced from the common documents the users visited after issuing those queries [10, 147, 156]. The second group expands queries with similar terms in the corresponding documents being visited in the history [32]. In addition to query expansion, click-through data has also been used to learn the rank function [70, 105].

More recently, several investigators have begun to analyze the temporal and dynamic nature of the click-through data [11, 27, 121, 144]. In [11], Beitzel *et al.* proposed the first approach to show the changes of popularities on an hourly basis. By using the categorization information of the Web queries, the results show that query traffic from particular topical categories differs from both the query stream as a whole and queries in other categories. Moreover, Shen *et al.* [121] proposed investigating the transitions among the topics of pages visited by a sample of Web search users. They constructed a model to predict the transitions in the topics for individual users and groups of users. Vlachos *et al.* [144] suggested identifying similar queries based on historical demand patterns, which are represented as time series using the best Fourier coefficients and the energy of the omitted components. Similarly, Chien and Immorlica [27] proposed finding semantically similar queries using the temporal correlation.

2.7 Collaborative Multimedia Retrieval

2.7.1 Image Retrieval

With the rapid growth of digital devices for capturing and storing multimedia data, multimedia information retrieval has become one of the most important research topics in recent years, among which image retrieval has been one of the key challenging problems. In the image retrieval field, content-based image retrieval (CBIR) is one of the most important topics, which has attracted a broad range of research interests in many computer science communities in the past decade [124]. Although extensive studies have been conducted, finding desired images from multimedia databases is still a challenging and open issue. The main challenge is due to the semantic gap between the low-level visual features extracted by computers and high-level human perception and interpretation [124]. Many early studies on CBIR focused primarily on low-level feature analysis [65, 126].

However, because of the complexity of visual image interpretation and the challenge of the semantic gap, it is impossible to discriminate all images by employing some rigid simple similarity measure on the low-level features. Although it is feasible to bridge the semantic gap by building an image index with textual descriptions, manual indexing on image databases is typically time-consuming, costly and subjective, and hence difficult to deploy fully in practical applications. Despite the promising results recently reported in image annotations [17, 66, 81], fully automatic image annotation is still a long way off. Relevance feedback, as an alternative and more feasible technique to mitigate the semantic gap issue, has been intensively investigated in recent years [111].

2.7.2 Relevance Feedback

Relevance feedback, originated from text-based information retrieval, is a powerful technique to improve retrieval performance [115]. In order to approach the query targets of an user, relevance feedback is viewed as the process of automatically altering an existing query by incorporating the relevance judgments that the user provided for previous retrieval tasks. In image retrieval, relevance feedback will first solicit the user's relevance judgments on the retrieved images returned by CBIR systems. Then, it refines retrieval results by learning the query targets from the relevance information provided. Although relevance feedback originated from text information retrieval, it is remarkable to see that later on it attracted much more attention in the field of image retrieval. In the past decade, various relevance feedback techniques have been proposed, ranging from heuristic methods to many sophisticated learning techniques [**?**, 61, 143].

The early relevance feedback for image retrieval was typically inspired by traditional relevance feedback in text retrieval. For example, Rui et al. [111] proposed learning according to the ranks of the positive and negative images along the feature axis in the feature space, which is similar to the idea of learning on “term frequency” and “inverse term frequency” in the text retrieval domain [106]. Later on, more systematic and comprehensive schemes were suggested to formulate the relevance feedback problem into an optimization problem. For example, MindReader formulated the feedback task as an optimization problem in which parameters are learned by minimizing the sum of overall distances from the query centroid to all relevant samples [62]. Rui et al. proposed a more rigorous approach called “Optimizing Learning”, which systematically formulates the relevance feedback as an optimizing problem and suggested a hierarchical learning approach rather than a flat model like the one in MindReader.

Recently, in parallel with the rapid developments in machine learning, a variety of machine learning techniques have been applied to the relevance feedback problem in image retrieval, including Bayesian learning [142], decision tree [89], boosting techniques [134], discriminant analysis [61, 169], dimension reduction [132, 169], ensemble learning [55, 133], etc. Moreover, some unsupervised learning techniques, like SOM [78] and EM algorithms [152], have also been evaluated in the literature. Recently, Support Vector Machines (SVMs) [141] have been widely explored in machine learning since they enjoy superior performance in real-world applications of pattern classification and recognition. Numerous investigations have applied SVMs to relevance feedback in CBIR [54, 60, 136, 162]. Previous studies have shown that SVM is one of the most promising and successful approaches for attacking the relevance feedback problem.

2.8 Convex Optimization

2.8.1 Overview of Convex Problems

Many machine learning problems studied in this thesis can be formulated as constrained optimization problems. These problems, with proper mathematical manipulations, can sometimes be expressed in convex form. These kinds of convex problems can be optimally solved very efficiently in practice. Specifically, interior-point methods are often used to solve these problems to a specified accuracy within a polynomial operations of the problem dimensions. More details about convex optimization theory can be found in reference [19].

Let us first look at some basic definitions of convex problems.

Definition 1 Convex Sets: *A set S is convex if the line segment between any two points in S lies in S, i.e., if for any $x_1, x_2 \in S$ and any θ with $0 \leq \theta \leq 1$, we have*

$$\theta x_1 + (1-\theta)x_2 \in S \ . \tag{2.21}$$

□

Definition 2 Convex Functions: *A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if **dom**f is a convex set and if for all $\mathbf{x}, \mathbf{y} \in \mathbf{dom}f$, and any θ with $0 \leq \theta \leq 1$, the following inequality holds:*

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}) \ . \tag{2.22}$$

□

Definition 3 Convex Problems: *A convex optimization problem (convex program) is defined as one being in the following form [19]:*

$$\min_{\mathbf{x}} \quad f_0(\mathbf{x}) \tag{2.23}$$

$$s.t. \quad f_i(\mathbf{x}) \leq 0 \qquad 1 \leq i \leq m, \tag{2.24}$$

$$h_i(\mathbf{x}) = 0 \qquad 1l \leq i \leq k, \tag{2.25}$$

where $\mathbf{x} \in \mathbb{R}^n$ *is the optimization variable,* $f_0, \ldots, f_m$ *are convex functions, and* $h_0, \ldots, h_k$ *are linear functions (affine functions).* □

In the above definition, the function f_0 is usually called the *objective function* or *cost function*. The inequalities are called *inequality constraints* and the equations are called *equality constraints*. If there is no constraint, the problem is an *unconstrained* problem. The subsequent parts review several types of convex optimization problems.

2.8.2 Linear Program

Definition 4 Linear Program (LP): *A convex optimization problem is called a linear program (LP) when the objective and constraint functions are all affine. The LP has the following general form:*

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & c^\top \mathbf{x} + d && (2.26)\\
s.t. \quad & G\mathbf{x} \preceq h, && (2.27)\\
& A\mathbf{x} = b, && (2.28)
\end{aligned}
$$

where $G \in \mathbb{R}^{m \times n}$ *and* $A \in \mathbb{R}^{p \times n}$. □

2.8.3 Quadratic Program

Definition 5 Quadratic Program (QP): *A convex optimization problem is called a quadratic program (QP) if the objective function is (convex) quadratic, and the constraint functions are affine. The QP is generally expressed as:*

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \tfrac{1}{2}\mathbf{x}^\top P \mathbf{x} + q^\top \mathbf{x} + r && (2.29)\\
s.t. \quad & G\mathbf{x} \preceq h, && (2.30)\\
& A\mathbf{x} = b, && (2.31)
\end{aligned}
$$

where $P \in \mathbb{S}^n_+$, $G \in \mathbb{R}^{m \times n}$ *and* $A \in \mathbb{R}^{p \times n}$. □

From the above definitions, one can see that quadratic programs include linear programs as a special case when $P = 0$.

2.8.4 Quadratically Constrained Quadratic Program

Definition 6 Quadratically Constrained Quadratic Program (QCQP): *A convex optimization problem is called a quadratically constrained quadratic program (QCQP) if the objective function and the constraint functions are all (convex) quadratic. The QCQP is generally expressed as:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \tfrac{1}{2}\mathbf{x}^\top P_0 \mathbf{x} + q_0^\top \mathbf{x} + r_0 && (2.32)\\ s.t. \quad & \tfrac{1}{2}\mathbf{x}^\top P_i \mathbf{x} + q_i^\top \mathbf{x} + r_i, \quad i = 1, \ldots, m && (2.33)\\ & A\mathbf{x} = b, && (2.34) \end{aligned}$$

where $P_i \in \mathbb{S}^n_+, i = 0, \ldots, m.$ □

It is evident that quadratically constrained quadratic programs include quadratic programs and linear programs as special cases.

2.8.5 Cone Program

In addition to convex optimization problems of standard forms, another type of very useful generalizations are convex optimization problems with generalized inequality constraints. One of the simplest case is the Cone Program (CP), which is defined as follows:

Definition 7 Cone Program (CP): *A convex optimization problem with generalized inequalities is called a Cone Program (CP) if the objective function is linear and the inequality constraint functions are affine:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & c^\top \mathbf{x} && (2.35)\\ s.t. \quad & F\mathbf{x} + g \preceq_K 0 && (2.36)\\ & A\mathbf{x} = b, && (2.37) \end{aligned}$$

where $K \subseteq \mathbb{R}^k$ *is a proper cone.* □

2.8.6 Semi-definite Program

Definition 8 Semi-definite Program (SDP): *A convex optimization problem with generalized inequalities is called a semi-definite program (SDP) if the objective function is linear and the inequality constraint functions are affine with the cone of positive semi-definite $k \times k$ matrices, i.e., K is $\mathbb{S}_+^k$. The SDP has the standard form as:*

$$\min_{\mathbf{x}} \quad c^\top \mathbf{x} \tag{2.38}$$

$$s.t. \quad \mathbf{x}_1 F_1 + \ldots + \mathbf{x}_n F_n + G \preceq 0 \tag{2.39}$$

$$A\mathbf{x} = b, \tag{2.40}$$

where G, $F_1, \ldots, F_n \in \mathbb{S}^k$ and $A \in \mathbb{R}^{p \times n}$. □

Similarly, SDP problems include LP, QP and QCQP as special cases. As a comparison of computational complexity, all of them can be solved efficiently in polynomial time. Among these four types of problems, in general, SDP is the hardest problem, QCQP is easier than SDP, QP is easier than QCQP, and LP is the easiest one.

□ **End of chapter.**

Chapter 3

Learning Unified Kernel Machines

3.1 Motivation

Classification is a core data mining technique and has been actively studied in the past decades. In general, the goal of classification is to assign unlabeled testing examples with a set of predefined categories. Traditional classification methods are usually conducted in a supervised learning way, in which only labeled data are used to train a predefined classification model. In the literature, a variety of statistical models have been proposed on classification in the machine learning and data mining communities. Amongst of the most popular and successful methodologies are the kernel-machine techniques, such as Support Vector Machines [141] and Kernel Logistic Regressions [171]. Like other early work for classification, traditional kernel-machine methods are usually performed in the supervised learning manner, which considers only the labeled data in the training phase.

Ideally, a good classification model should take advantage of not only the labeled data, but also the unlabeled data when they are available. Learning on both labeled and unlabeled data has become an important research topic in recent years. One way to exploit the unlabeled data is to use active learning [29]. The goal of active learning is to choose the most informative example from the unlabeled data for manual labeling. In recent years, active learning has been applied to many classification tasks [85].

Another popular emerging technique for exploiting unlabeled data is semi-supervised learning [26], which has attracted a surge of research attention recently [170]. A variety of machine-learning techniques have been proposed for semi-supervised learning, of which the most well-known approaches are based on the graph Laplacians methodology [166, 174, 26]. While promising results have been generally reported in this research area, there are so far few comprehensive semi-supervised learning schemes applicable to large-scale classification problems.

Although supervised learning, semi-supervised learning and active learning have been studied separately, so far there are few comprehensive schemes to combine these techniques together effectively for classification tasks. To this end, we propose a general framework of learning the Unified Kernel Machines (UKM) [24, 25] by unifying supervised kernel-machine learning, semi-supervised learning, unsupervised kernel design and active learning together for large-scale classification problems.

The rest of this chapter is organized as follows. Section 3.2 presents our framework for learning the unified kernel machines. Section 3.3 proposes a new Spectral Kernel Learning (SKL) algorithm for learning semi-supervised kernels. Section 3.4 presents a specific UKM paradigm for classification, namely the Unified Kernel Logistic Regression (UKLR). Section 3.5 evaluates the empirical performance of our proposed algorithm and the UKLR classification scheme. Section 3.6 analyzes the complexity and scalability of our proposed method. Section 3.7 summarizes this chapter.

3.2 Unified Kernel Machines Framework

In this section, we present the framework for learning the unified kernel machines by combining supervised kernel machines, semi-supervised kernel learning and active learning techniques into a unified solution. Figure 3.1 gives an overview of our proposed scheme. For simplicity, we restrict our discussions to classification problems.

Let $\mathcal{M}(K, \alpha)$ denote a kernel machine that has some underlying probabilistic model, such as a kernel logistic regressions (or a support vector machine). In general, a kernel machine contains two components, i.e., the kernel K (either a kernel function or simply a kernel matrix), and the model parameters α. In traditional supervised kernel-machine learning, the kernel K is usually a known parametric kernel function and the goal of the learning task is usually to determine the model parameter α. This often limits the performance of the kernel machine if the specified kernel is not appropriate.

To this end, we propose a unified scheme to learn the unified kernel machines by learning on both the kernel K and the model parameters α together. In order to exploit the unlabeled data, we suggest combining semi-supervised kernel learning and active learning techniques together for learning the unified kernel machines effectively from the labeled and unlabeled data. More specifically, we outline a general framework of learning the unified kernel machine as follows.

Let L denote the labeled data and U denote the unlabeled data. The goal of the unified kernel machine learning task is to learn the kernel machine $\mathcal{M}(K^*, \alpha^*)$ that can classify the data effectively. Specifically, it includes the following five steps:

- **Step 1. Kernel Initialization**

  The first step is to initialize the kernel component K_0 of the kernel machine $\mathcal{M}(K_0, \alpha_0)$. Typically, users can specify the initial ker-

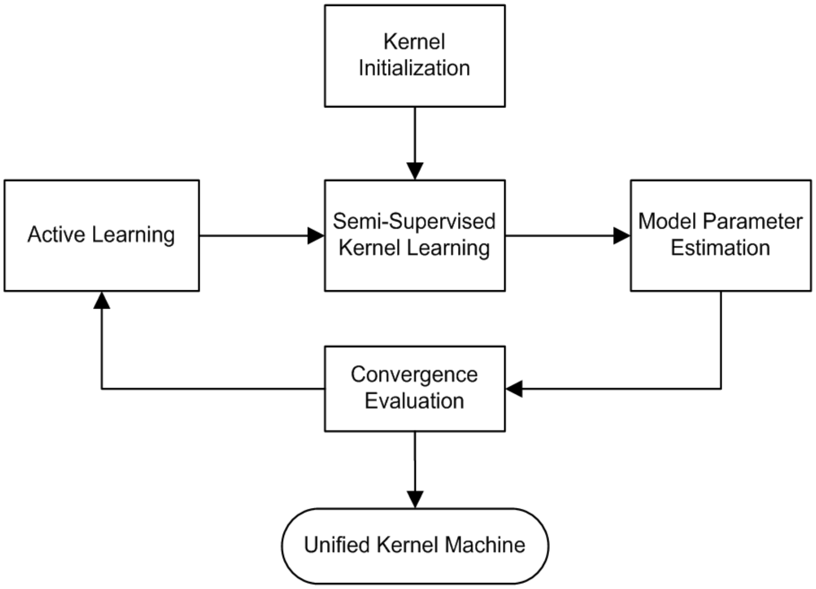


Figure 3.1: Learning the unified kernel machines

nel K_0 (function or matrix) with a standard kernel. When some domain knowledge is available, users can also design a kernel with domain knowledge (or some data-dependent kernels).

- **Step 2. Semi-Supervised Kernel Learning**

  The initial kernel may not be good enough to classify the data correctly. Hence, we propose employing the semi-supervised kernel learning technique to learn a new kernel K by engaging both the labeled L and unlabeled data U available.

- **Step 3. Model Parameter Estimation**

  When the kernel K is known, to estimate the parameters of the kernel machines based on some model assumption, such as Kernel Logistic Regression or Support Vector Machines, one can simply employ the standard supervised kernel-machine learning approach to solve the model parameters α.

- **Step 4. Active Learning**

  In many classification tasks, labeling is expensive. Active learning is an important way of reducing human effort involved in labeling. Typically, we can choose a batch of the most informative examples S that can most effectively update the current kernel machine $\mathcal{M}(K, \alpha)$.

- **Step 5. Convergence Evaluation**

  The last step is the convergence evaluation, in which we check whether the kernel machine is good enough for the classification task. If it is not, we will repeat the above steps until a satisfactory kernel machine is obtained.

This is a general framework for learning unified kernel machines. In this chapter, we focus our main attention on a semi-supervised kernel learning technique, which is a core component of learning the unified kernel machines.

3.3 Spectral Kernel Learning

We propose a new semi-supervised kernel learning method, which is a fast and robust algorithm for learning semi-supervised kernels from labeled and unlabeled data. In the following parts, we first introduce the theoretical foundation and then present our spectral kernel learning algorithm. Finally, we show the connections of our method to existing work and justify the effectiveness of our solution from empirical observations.

3.3.1 Theoretical Foundation

Let us first consider a standard supervised kernel learning problem. Assume that the data (X, Y) are drawn from an unknown distribution

$\mathcal{D}$. The goal of supervised learning is to find a prediction function $p(\mathbf{X})$ that minimizes the following expected true loss:

$$E_{(X,Y)\sim\mathcal{D}}\mathcal{L}(p(X), Y),$$

where $E_{(X,Y)\sim\mathcal{D}}$ denotes the expectation over the true underlying distribution $\mathcal{D}$. In order to achieve a stable estimation, we usually need to restrict the size of hypothesis function family. Given l training examples $(\mathbf{x}_1,y_1)$,. . .,$(\mathbf{x}_l,y_l)$, we typically train a prediction function $\hat{p}$ in a reproducing Hilbert space $\mathcal{H}$ by minimizing the empirical loss [141]. Since the reproducing Hilbert space can be large, to avoid overfitting problems, we often consider a regularized method as follows:

$$\hat{p} = \arg\inf_{p\in\mathcal{H}} \left(\frac{1}{l}\sum_{i=1}^{l} \mathcal{L}(p(\mathbf{x}_i), y_i) + \lambda ||p||_{\mathcal{H}}^2 \right) , \tag{3.1}$$

where λ is a chosen positive regularization parameter. It can be shown that the solution of (3.1) can be represented as the following kernel method:

$$\hat{p}(\mathbf{x}) = \sum_{i=1}^{l} \hat{\alpha}_i k(\mathbf{x}_i, \mathbf{x})$$

$$\boldsymbol{\alpha} = \arg\inf_{\boldsymbol{\alpha}\in\mathbb{R}^l} \left(\frac{1}{l}\sum_{i=1}^{l} \mathcal{L}\left(p(\mathbf{x}_i), y_i\right) + \lambda \sum_{i,j=1}^{l} \alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \right) ,$$

where $\boldsymbol{\alpha}$ is a parameter vector to be estimated from the data and k is a kernel, which is known as *kernel function*. Typically a *kernel* returns the inner product between the mapping images of two given data examples, such that $k(\mathbf{x}_i, \mathbf{x}_j) = \langle\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$ for $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$.

Let us now consider a semi-supervised learning setting. Given labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{n}$, we consider learning the real-valued vectors $f \in \mathbb{R}^n$ by the following semi-supervised learning method:

$$\hat{f} = \arg\inf_{f\in\mathbb{R}^n} \left(\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f_i, y_i) + \lambda f^\top K^{-1} f \right) , \tag{3.2}$$

where K is an $m \times m$ kernel matrix with $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. Zhang et al. [163] proved that the solution of the above semi-supervised learning is equivelent to the solution of standard supervised learning in (3.1), such that

$$\hat{f}_j = \hat{p}(\mathbf{x}_j) \quad j = 1, \ldots, m. \tag{3.3}$$

The theorem offers a principle of unsupervised kernel design: one can design a new kernel $\bar{k}(\cdot, \cdot)$ based on the unlabeled data and then replace the original kernel k by $\bar{k}$ in the standard supervised kernel learning. More specifically, the framework of spectral kernel design suggests that the new kernel matrix $\bar{K}$ can be designed by means of a function g as follows:

$$\bar{K} = \sum_{i=1}^{n} g(\lambda_i) \mathbf{v}_i \mathbf{v}_i^{\top} , \tag{3.4}$$

where $(\lambda_i, \mathbf{v}_i)$ are the eigen-pairs of the original kernel matrix K, and the function $g(\cdot)$ can be regarded as a filter function or a transformation function that modifies the spectra of the kernel. The authors in [163] showed a theoretical justification that designing a kernel matrix with faster spectral decay rates should result in better generalization performance, which offers an important principle for learning an effective kernel matrix.

On the other hand, there are some recent papers that have studied theoretical principles for learning effective kernel functions or matrices from labeled and unlabeled data. One important work is the *kernel target alignment*, which can be used not only to assess the relationship between the feature spaces of two kernels, but also to measure the similarity between the feature space of a kernel and the feature space induced by labels [31]. Specifically, given two kernel matrices K_1 and K_2, their relationship is defined in terms of the following score of *alignment*:

Definition 9 Kernel Alignment: *The empirical alignment of two* *given kernels K_1 and K_2 with respect to the sample set S is the quantity:*

$$\hat{A}(K_1, K_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}} \tag{3.5}$$

where K_i is the kernel matrix induced by the kernel k_i and $\langle \cdot, \cdot \rangle$ is the Frobenius product between two matrices, i.e.,
$\langle K_1, K_2 \rangle_F = \sum_{i,j=1}^{n} k_1(\mathbf{x}_i, \mathbf{x}_j) k_2(\mathbf{x}_i, \mathbf{x}_j)$. □

The above definition of *kernel alignment* offers a means of learning the kernel matrix by assessing the relationship between a given kernel and a *target kernel* induced by the given labels. Let $\mathbf{y} = \{y_i\}_{i=1}^{l}$ denote a vector of labels in which $y_i \in \{+1, -1\}$ for binary classification. Then the target kernel can be defined as $T = \mathbf{y}\mathbf{y}^\top$. Let K be the kernel matrix with the following structure

$$K = \begin{pmatrix} K_{tr} & K_{trt} \\ K_{trt}^\top & K_t \end{pmatrix} \tag{3.6}$$

where $K_{ij} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, K_{tr} denotes the matrix part of the "training-data block" and K_t denotes the matrix part of the "test-data block."

The theory in [31] provides the principle of learning the kernel matrix, i.e., looking for a kernel matrix K with good generalization performance is equivalent to finding the matrix that maximizes the following empirical kernel alignment score:

$$\hat{A}(K_{tr}, T) = \frac{\langle K_{tr}, T \rangle_F}{\sqrt{\langle K_{tr}, K_{tr} \rangle_F \langle T, T \rangle_F}} \tag{3.7}$$

This principle has been used to learn kernel matrices with multiple kernel combinations [80] and also semi-supervised kernels from graph Laplacians [175]. Motivated by the related theoretical work, we propose a new spectral kernel learning (SKL) algorithm, which learns spectra of the kernel matrix by obeying both the principle of unsupervised kernel design and the principle of kernel target alignment.

3.3.2 Algorithm

Assume that we are given a set of labeled data $L = \{\mathbf{x}_i, y_i\}_{i=1}^{l}$, a set of unlabeled data $U = \{\mathbf{x}_i\}_{i=l+1}^{n}$, and an initial kernel matrix K. We first conduct the eigen-decomposition of the kernel matrix:

$$K = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top \ , \tag{3.8}$$

where $(\lambda_i, \mathbf{v}_i)$ are eigen pairs of K and are assumed to be in decreasing order, i.e., $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. For efficiency considerations, we select the top d eigen pairs, such that

$$K_d = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top \approx K \ , \tag{3.9}$$

where the parameter $d \ll n$ is a dimension cutoff factor that can be determined by some criterion, such as the cumulative eigen energy.

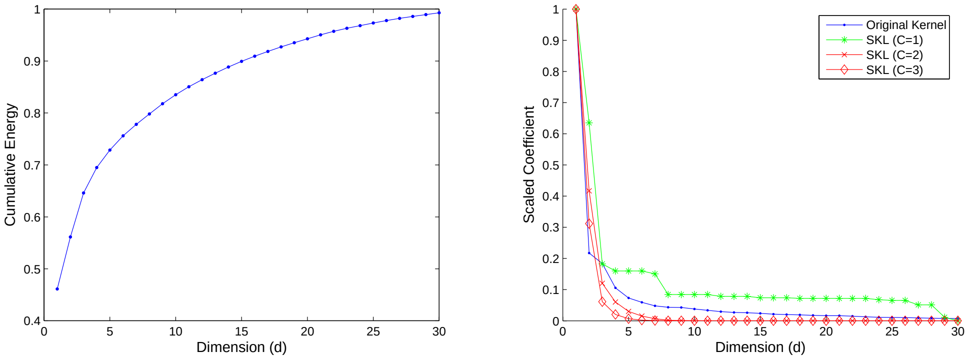
Based on the principle of unsupervised kernel design, we proceed to learn the kernel matrix as follows:

$$\bar{K} = \sum_{i=1}^{d} \mu_i \mathbf{v}_i \mathbf{v}_i^\top \ , \tag{3.10}$$

where $\mu_i \geq 0$ are spectral coefficients of the new kernel matrix. The goal of the Spectral Kernel Learning (SKL) algorithm is to find the optimal spectral coefficients μ_i for the following optimization:

$$\begin{array}{rc} \max\limits_{\bar{K},\mu} & \hat{A}(\bar{K}_{tr}, T) \\ \text{subject to} & \bar{K} = \sum_{i=1}^{d} \mu_i \mathbf{v}_i \mathbf{v}_i^\top \\ & trace(\bar{K}) = 1 \\ & \mu_i \geq 0, \\ & \mu_i \geq C\mu_{i+1}, i = 1 \ldots d-1 \ , \end{array} \tag{3.11}$$

where C is a decay factor that satisfies $C \geq 1$, $\mathbf{v}_i$ are the top d eigen vectors of the original kernel matrix K, $\bar{K}_{tr}$ is the kernel matrix restricted to the (labeled) training data and T is the target kernel induced by the labels. Note that C is introduced as an important parameter to control the decay rate of spectral coefficients; this will influence the overall performance of the kernel machine.



(a) Cumulative eigen energy

(b) Spectral coefficients

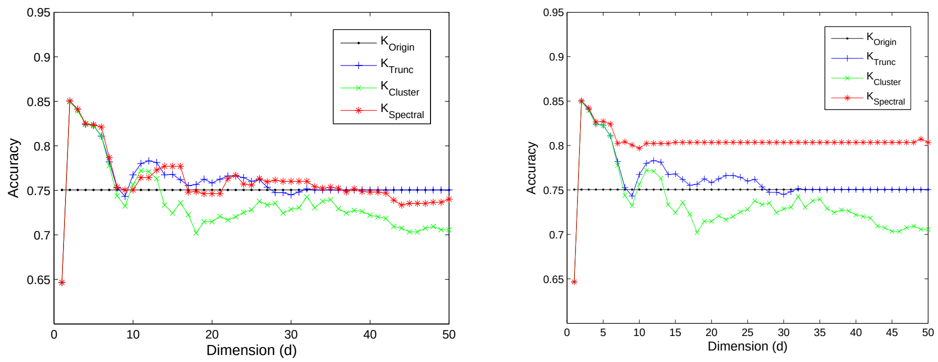
Figure 3.2: Illustration of cumulative eigen energy and the spectral coefficients of different decay factors on the Ionosphere dataset. The initial kernel is a linear kernel and the number of labeled data is 20.

The above optimization problem is a convex optimization and is often regarded as a general cone program [80], which may not be computationally efficient. In the following, we turn it into a QP problem that can be solved more efficiently.

Because the objective function (3.7) is invariant to scales, we can remove the constant term $\langle T, T\rangle_F$ from the objective function, which results in the following form:

$$\frac{\langle \bar{K}_{tr}, T\rangle_F}{\sqrt{\langle \bar{K}_{tr}, \bar{K}_{tr}\rangle_F}} \,. \tag{3.12}$$

In order to remove the trace constraint in (3.11), we consider the following alternative approach. Instead of maximizing the objective function (3.12) directly, we can set the numerator to 1 and then minimize the



(a) C=1

(b) C=2

Figure 3.3: Classification performance of semi-supervised kernels with different decay factors on the Ionosphere dataset. The initial kernel is a linear kernel and the number of labeled data is 20.

denominator. Therefore, we can turn the optimization problem into:

$$\begin{aligned} \min_{\mu} \quad & \sqrt{\langle \bar{K}_{tr}, \bar{K}_{tr} \rangle_F} \\ \text{subject to} \quad & \bar{K} = \textstyle\sum_{i=1}^{d} \mu_i \mathbf{v}_i \mathbf{v}_i^{\top} \\ & \langle \bar{K}_{tr}, T \rangle_F = 1 \\ & \mu_i \geq 0, \\ & \mu_i \geq C\mu_{i+1}, i = 1 \ldots d-1 \ . \end{aligned} \tag{3.13}$$

This minimization problem without the trace constraint is equivalent to the original maximization problem with the trace constraint.

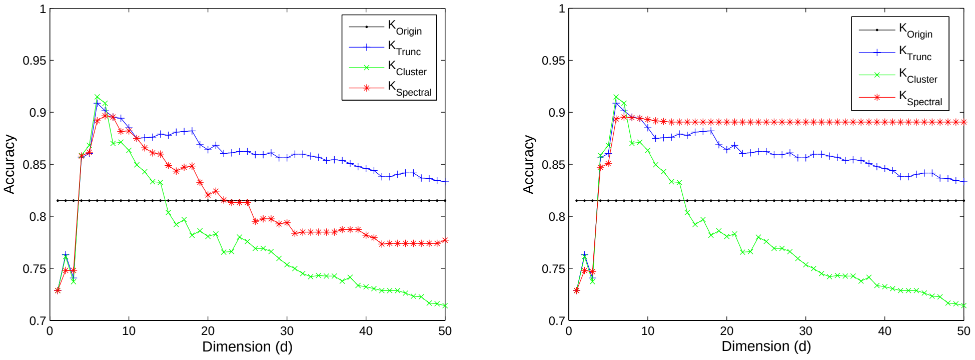
Let $vec(A)$ denote the column vectorization of a matrix A and let $D = [vec(V_{1,tr}) \ldots vec(V_{d,tr})]$ be a constant matrix of size $l^2 \times d$, in which the d matrices of $V_i = \mathbf{v}_i \mathbf{v}_i^{\top}$ are with size of $l \times l$. It is not difficult to show that the above problem is equivalent to the following optimization:

$$\begin{aligned} \min_{\mu} \quad & ||D\mu|| \\ \text{subject to} \quad & vec(T)^{\top} D\mu = 1 \\ & \mu_i \geq 0 \\ & \mu_i \geq C\mu_{i+1}, i = 1 \dots d-1 \ . \end{aligned} \tag{3.14}$$

Minimizing the norm is then equivalent to minimizing the squared norm. Hence, we can obtain the final optimization problem as:

$$\begin{aligned} \min_{\mu} \quad & \mu^{\top} D^{\top} D\mu \\ \text{subject to} \quad & vec(T)^{\top} D\mu = 1 \\ & \mu_i \geq 0 \\ & \mu_i \geq C\mu_{i+1}, i = 1 \dots d-1 \ . \end{aligned}$$

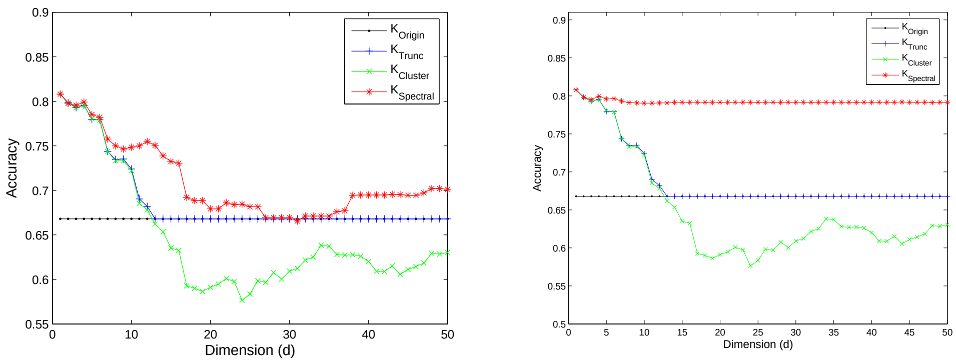
This is a standard QP problem that can be solved efficiently.



(a) C=1

(b) C=2

Figure 3.4: Classification performance of spectral kernel learning methods with different decay factors on the Ionosphere dataset. The initial kernel is an RBF kernel and the number of labeled data is 20.



(a) C=1

(b) C=2

Figure 3.5: Classification performance of semi-supervised kernels with different decay factors on the Heart dataset. The initial kernel is a linear kernel and the number of labeled data is 20.

3.3.3 Connections and Justifications

The essence of our semi-supervised kernel learning method is based on the theories of unsupervised kernel design and kernel target alignment. More specifically, we consider a dimension-reduction effective method to learn the semi-supervised kernel that maximizes the kernel alignment score. By examining previous work on unsupervised kernel design, the following two situations can be summarized as special cases of the SKL framework:

- **Cluster Kernel**

  This method adopts a "[1. . . ,1,0,. . . ,0]" kernel that has been used in spectral clustering [98]. It sets the largest spectral coefficients to 1 and the rest to 0, i.e.,

$$\mu_i = \begin{cases} 1 & \text{for} \quad i \leq d \\ 0 & \text{for} \quad i > d \end{cases} . \tag{3.15}$$

For comparison, we refer to this method as "Cluster kernel" denoted by $K_{\texttt{Cluster}}$.

- **Truncated Kernel**

  Another method is called the truncated kernel, which keeps only the top d largest spectral coefficients:

$$\mu_i = \begin{cases} \lambda_i & \text{for} \quad i \leq d \\ 0 & \text{for} \quad i > d \end{cases} , \tag{3.16}$$

  where λ_i are the top eigen values of an initial kernel. We can see that this is exactly equivalent to the method of kernel principal component analysis [117] that keeps only the d most significant principal components of a given kernel. For a comparison, we denote this method as $K_{\texttt{Trunc}}$.

In our case, in comparison with semi-supervised kernel learning methods by graph Laplacians, our work is similar to the approach in [175], which learns the spectral transformation of graph Laplacians by kernel target alignment with order constraints. However, we should emphasize two important differences that will explain why our method can work more effectively.

First, the work in [175] represents traditional graph based semi-supervised learning methods, which assume the kernel matrix is derived from the spectral decomposition of graph Laplacians. Instead, our spectral kernel learning method learns on any initial kernel and assumes the kernel matrix is derived from the spectral decomposition of the normalized kernel.

Second, compared to the kernel learning method in [80], the authors in [175] proposed the addition of order constraints into the optimization of the kernel target alignment [31] to enforce the constraints of graph smoothness. In our case, we suggest a decay factor C to constrain the relationship of spectral coefficients in the optimization in order to make the spectral coefficients decay faster. In fact, if we ignore the difference of graph Laplacians and assume that the initial kernel in our method is given as $K \approx L^{-1}$, we can see that the method in [175] can be regarded as a special case of our method when the decay factor C is set to 1 and the dimension cut-off parameter d is set to n.

3.3.4 Empirical Observations

Setting $C = 1$ in the spectral kernel learning algorithm may not be a good choice for learning an effective kernel. To support this statement, let us give some empirical examples to justify the design of the spectral kernel learning algorithm. One goal of the spectral kernel learning methodology is to attain a fast decay rate of the spectral coefficients of the kernel matrix. Figure 3.2 illustrates an example of the change in the resulting spectral coefficients using different decay factors in our spectral kernel learning algorithms. From the figure, we can see that the curves with larger decay factors ($C = 2, 3$) have faster decay rates than the original kernel and the one using $C = 1$. Meanwhile, we can see that the cumulative eigen energy score quickly converges to 100% when the number of dimensions is increased. This shows that we may use much smaller numbers of eigen-pairs in our semi-supervised kernel learning algorithm for large-scale problems.

To examine the impact of different decay factors in more details, we evaluate the classification performance of spectral kernel learning methods with different decay factors in Figure 3.3. We can see that two unsupervised kernels, $K_{\texttt{Trunc}}$ and $K_{\texttt{Cluster}}$, tend to perform better than the original kernel when the dimension is small. But their performances are not very stable when the number of dimensions is increased. For comparison, the spectral kernel learning method achieves very stable and good performance when the decay factor C is larger than 1. When the decay factor is equal to 1, the performance becomes unstable due to the slow decay rates observed from our previous results in Figure 3.3. This observation matches the theoretical justification [163] that a kernel with good performance usually favors a faster decay rate of spectral coefficients.

Figure 3.4 and Figure 3.5 illustrate more empirical examples based on different initial kernels, in which similar results can be observed. Note that our suggested kernel learning method can learn on any valid kernel, and different initial kernels will impact the performance of the resulting spectral kernels. It is usually helpful if the initial kernel is provided with domain knowledge.

3.4 Unified Kernel Logistic Regression

In this section, we present a specific paradigm based on the proposed framework of learning unified kernel machines. We assume the underlying probabilistic model of the kernel machine is Kernel Logistic Regression (KLR). Based on the UKM framework, we develop the Unified Kernel Logistic Regression (UKLR) paradigm to tackle classification tasks. Note that our framework is not restricted to the KLR model, but also can be widely extended for many other kernel machines, such as Support Vector Machine (SVM) and Regularized Least-Square (RLS) classifiers.

Similar to other kernel machines, such as SVM, a KLR problem can be formulated in terms of a standard regularized form of *loss+penalty* in the reproducing kernel Hilbert space (RKHS):

$$\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} \ln(1 + e^{-y_i f(\mathbf{x}_i)}) + \frac{\lambda}{2} ||f||^2_{\mathcal{H}_K} \ , \tag{3.17}$$

where $\mathcal{H}_K$ is the RKHS by a kernel K and λ is a regularization parameter. By the representer theorem, the optimal $f(\mathbf{x})$ has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \ , \tag{3.18}$$

where α_i are model parameters. Note that we omit the constant term in $f(x)$ for simplified notations. To solve the KLR model parameters, there are a number of available techniques for effective solutions [171].

```
Algorithm: Unified Kernel Logistic Regresssion
Input
    • $K_0$: Initial normalized kernel
    • $L$: Set of labeled data
    • $U$: Set of unlabeled data
Repeat
    • Spectral Kernel Learning
      $K \leftarrow$ Spectral_Kernel($K_0$, $L$, $U$);
    • KLR Parameter Estimation
      $\alpha \leftarrow$ KLR_Solver($L$, $K$);
    • Convergence Test
      If (converged), Exit Loop;
    • Active Learning
      $\mathbf{x}^* \leftarrow max_{\mathbf{x} \in U}\ H(\mathbf{x}; \alpha, K)$
      $L \leftarrow L \cup \{\mathbf{x}^*\}$, $U \leftarrow U - \{\mathbf{x}^*\}$
Until converged.
Output
    • UKLR = $\mathcal{M}(K, \alpha)$.
```

Figure 3.6: The unified kernel logistic regression algorithm.

When the kernel K and the model parameters α are available, we use the following solution for active learning, which is simple and efficient for large-scale problems. More specifically, we measure the information entropy of each unlabeled data example as follows

$$H(\mathbf{x}; \alpha, K) = -\sum_{i=1}^{N_C} p(C_i|\mathbf{x}) log(p(C_i|\mathbf{x})) \ , \tag{3.19}$$

where N_C is the number of classes and C_i denotes the i^{th} class and

$p(C_i|\mathbf{x})$ is the probability of the data example $\mathbf{x}$ belonging to the i^{th} class which can be naturally obtained by the current KLR model (α, K). The unlabeled data examples with maximum values of entropy will be considered as the most informative data for labeling.

By unifying the spectral kernel learning method proposed in Section 3, we summarize the proposed algorithm of Unified Kernel Logistic Regression (UKLR) in Figure 3.6. In the algorithm, note that we can usually initialize a kernel by a standard kernel with appropriate parameters determined by cross validation or by a proper design of the initial kernel with domain knowledge.

3.5 Experimental Results

We discuss our empirical evaluation of the proposed framework and algorithms for classification. We first evaluate the effectiveness of our suggested spectral kernel learning algorithm for learning semi-supervised kernels and then compare the performance of our unified kernel logistic regression paradigm with traditional classification schemes.

3.5.1 Experimental Testbed and Settings

We use the datasets from UCI machine learning repository[1]. Four datasets are employed in our experiments. Table 3.1 shows the details of four UCI datasets in our experiments.

For experimental settings, to examine the influences of different training sizes, we test the compared algorithms on four different training set sizes for each of the four UCI datasets. For each given training set size, we conduct 100 random trials in which a labeled set is randomly sampled from the whole dataset and all classes must be present in the sampled labeled set. The rest data examples of the dataset are

[1] www.ics.uci.edu/ mlearn/MLRepository.html

Table 3.1: List of UCI machine learning datasets.

Dataset	#Instances	#Features	#Classes
Heart	270	13	2
Ionosphere	351	34	2
Sonar	208	60	2
Wine	178	13	3

then used as the testing (unlabeled) data. To train a classifier, we employ the standard KLR model for classification. We choose the bounds on the regularization parameters via cross validation for all compared kernels to avoid an unfair comparison. For multi-class classification, we perform one-against-all binary training and testing and then pick the class with the maximum class probability.

3.5.2 Semi-Supervised Kernel Learning

In this part, we evaluate the performance of our spectral kernel learning algorithm for learning semi-supervised kernels. We implemented our algorithm by a standard Matlab Quadratic Programming solver (quadprog). The dimension-cut parameter d in our algorithm is fixed to 20 without further optimizing. Note that one can easily determine an appropriate value of d by examining the range of the cumulative eigen energy score in order to reduce the computational cost for large-scale problems. The decay factor C is important for our spectral kernel learning algorithm. As we have shown examples before, C must be a positive real value greater than 1. Typically we favor a larger decay factor to achieve better performance. But it must not be set too large since the too large decay factor may result in the overly stringent constraints in the optimization which gives no solutions. In our experiments, C is fixed to constant values (greater than 1) for the engaged datasets.

For a comparison, we compare our SKL algorithms with the state-of-the-art semi-supervised kernel learning method by graph Laplacians [175], which is related to a QCQP problem. More specifically, we have implemented two graph Laplacians based semi-supervised kernels by order constraints [175]. One is the order-constrained graph kernel (denoted as "Order") and the other is the improved order-constrained graph kernel (denoted as "Imp-Order"), which removes the constraints from constant eigenvectors. To carry a fair comparison, we use the top 20 smallest eigenvalues and eigenvectors from the graph Laplacian which is constructed with 10-NN unweighted graphs. We also include three standard kernels for comparisons.

Table 3.2 shows the experimental results of the compared kernels (3 standard and 5 semi-supervised kernels) based on KLR classifiers on four UCI datasets with different sizes of labeled data. The mean accuracies and standard errors are shown in the table. Each cell in the table has two rows: the upper row shows the average testing set accuracies with standard errors; and the lower row gives the average run time in *seconds* for learning the semi-supervised kernels on a 3GHz desktop computer ("Order" and "Imp-Order" kernels are solved by SeDuMi/YALMIP package; "SKL" kernels are solved directly by the Matlab quadprog function. We conducted a paired t-test at significance level of 0.05 to assess the statistical significance of the test set accuracy results. From the experimental results, we found that the two order-constrained based graph kernels perform well in the *Ionosphere* and *Wine* datasets, but they do not achieve important improvements on the *Heart* and *Sonar* datasets. Among all the compared kernels, the semi-supervised kernels by our spectral kernel learning algorithms achieve the best performances. The semi-supervised kernel initialized with an RBF kernel outperforms other kernels in most cases. For example, in *Ionosphere* dataset, an RBF kernel with 10 initial training examples only achieves 73.56% test set accuracy, and the SKL algorithm can boost the accuracy significantly to 83.36%. Finally, looking into the time performance, the average run time of our algorithm is less than 10% of the previous QCQP algorithms.

3.5.3 Unified Kernel Logistic Regression

In this part, we evaluate the performance of our proposed paradigm of unified kernel logistic regression (UKLR). As a comparison, we implement two traditional classification schemes: one is traditional KLR classification scheme that is trained on randomly sampled labeled data, denoted as "KLR+Rand." The other is the active KLR classification scheme that actively selects the most informative examples for labeling, denoted as "KLR+Active." The active learning strategy is based on a simple maximum entropy criteria given in the pervious section. The UKLR scheme is implemented based on the algorithm in Figure 3.6.

For active learning evaluation, we choose a batch of 10 most informative unlabeled examples for labeling in each trial of evaluations. Table 3.3 summarizes the experimental results of average test set accuracy performances on four UCI datasets. From the experimental results, we can observe that the active learning classification schemes outperform the randomly sampled classification schemes in most cases. This shows the suggested simple active learning strategy is effectiveness. Further, among all compared schemes, the suggested UKLR solution significantly outperforms other classification approaches in most cases. These results show that the unified scheme is effective and promising to integrate traditional learning methods together in a unified solution.

3.6 Computational Complexity and Scalability

In this section, let us analyze the time complexity of our proposed algorithms in the framework of learning unified kernel machines. Since

Table 3.2: Classification performance of different kernels using KLR classifiers on four datasets.

Train	**Standard Kernels**			**Semi-Supervised Kernels**				
Size	Linear	Quadratic	RBF	Order	Imp-Order	SKL(Linear)	SKL(Quad)	SKL(RBF)
Heart								
10	66.92 ± 0.91 —	71.33 ± 0.82 —	70.76 ± 0.82 —	63.39 ± 0.76 (0.74)	63.39 ± 0.76 (0.65)	70.69 ± 0.86 (0.08)	71.49 ± 0.80 (0.08)	**72.55 ± 0.77** (0.08)
20	70.79 ± 0.66 —	71.30 ± 0.67 —	73.74 ± 0.54 —	69.68 ± 0.75 (0.95)	69.68 ± 0.75 (0.85)	76.82 ± 0.40 (0.08)	**77.12 ± 0.44** (0.08)	76.78 ± 0.46 (0.08)
30	75.35 ± 0.47 —	70.98 ± 0.49 —	75.30 ± 0.43 —	72.50 ± 0.52 (0.91)	72.50 ± 0.52 (0.89)	78.27 ± 0.37 (0.08)	78.67 ± 0.37 (0.08)	**78.70 ± 0.41** (0.08)
40	77.41 ± 0.38 —	70.47 ± 0.50 —	76.22 ± 0.37 —	74.57 ± 0.45 (1.12)	74.57 ± 0.45 (1.09)	78.91 ± 0.33 (0.08)	**79.67 ± 0.33** (0.08)	79.45 ± 0.32 (0.08)
Ionosphere								
10	72.35 ± 0.66 —	69.96 ± 0.65 —	74.12 ± 0.71 —	71.46 ± 1.33 (0.89)	71.46 ± 1.33 (0.78)	72.82 ± 0.92 (0.10)	70.04 ± 0.86 (0.10)	**81.17 ± 0.61** (0.10)
20	75.41 ± 0.50 —	75.54 ± 0.75 —	81.64 ± 0.71 —	83.77 ± 0.81 (0.80)	83.77 ± 0.81 (0.77)	79.95 ± 0.48 (0.09)	78.88 ± 0.74 (0.09)	**88.17 ± 0.57** (0.09)
30	76.97 ± 0.36 —	80.02 ± 0.66 —	87.02 ± 0.48 —	87.88 ± 0.52 (0.98)	87.88 ± 0.52 (0.94)	81.80 ± 0.37 (0.09)	81.76 ± 0.75 (0.09)	**90.12 ± 0.45** (0.08)
40	77.63 ± 0.30 —	82.34 ± 0.53 —	89.38 ± 0.36 —	90.11 ± 0.30 (1.18)	90.11 ± 0.30 (1.14)	83.35 ± 0.27 (0.09)	84.79 ± 0.58 (0.09)	**90.40 ± 0.39** (0.08)
Sonar								
10	63.71 ± 0.54 —	63.85 ± 0.51 —	61.24 ± 0.64 —	59.23 ± 0.54 (0.66)	59.23 ± 0.54 (0.67)	63.70 ± 0.73 (0.09)	62.67 ± 0.82 (0.09)	**66.14 ± 0.71** (0.09)
20	65.23 ± 0.54 —	67.64 ± 0.57 —	64.72 ± 0.65 —	64.03 ± 0.55 (0.76)	64.03 ± 0.55 (0.72)	67.68 ± 0.57 (0.09)	65.40 ± 0.62 (0.09)	**68.96 ± 0.54** (0.09)
30	66.34 ± 0.52 —	70.52 ± 0.46 —	68.33 ± 0.55 —	67.26 ± 0.50 (0.90)	67.26 ± 0.50 (0.88)	71.07 ± 0.46 (0.08)	69.42 ± 0.46 (0.08)	**71.80 ± 0.45** (0.08)
40	66.03 ± 0.50 —	71.45 ± 0.46 —	69.42 ± 0.56 —	68.69 ± 0.42 (1.38)	68.69 ± 0.42 (1.22)	72.12 ± 0.43 (0.10)	70.42 ± 0.43 (0.10)	**72.46 ± 0.45** (0.10)
Wine								
10	81.97 ± 0.79 —	85.45 ± 0.65 —	88.77 ± 0.63 —	89.55 ± 0.80 (1.66)	89.55 ± 0.80 (1.41)	86.64 ± 0.74 (0.16)	88.07 ± 0.83 (0.14)	**93.45 ± 0.42** (0.14)
20	86.28 ± 0.69 —	85.85 ± 0.60 —	93.67 ± 0.41 —	91.72 ± 0.59 (0.67)	91.72 ± 0.59 (0.69)	92.49 ± 0.47 (0.07)	92.57 ± 0.67 (0.08)	**95.32 ± 0.44** (0.07)
30	93.08 ± 0.30 —	87.34 ± 0.52 —	94.34 ± 0.29 —	94.59 ± 0.31 (0.97)	94.59 ± 0.31 (1.09)	94.60 ± 0.23 (0.08)	94.58 ± 0.28 (0.08)	**96.41 ± 0.17** (0.08)
40	95.52 ± 0.24 —	88.73 ± 0.40 —	96.06 ± 0.14 —	96.12 ± 0.20 (1.03)	96.12 ± 0.20 (1.22)	96.08 ± 0.17 (0.07)	95.90 ± 0.16 (0.07)	**97.12 ± 0.13** (0.07)

Table 3.3: Classification performance of different classification schemes on four UCI datasets. The mean accuracies and standard errors are shown in the table. "KLR" represents the initial classifier with the initial train size; other three methods are trained with additional 10 random/active examples.

Train	**Linear Kernel**				**RBF Kernel**			
Size	KLR	KLR+Rand	KLR+Active	UKLR	KLR	KLR+Rand	KLR+Active	UKLR
Heart								
10	66.92 ± 0.91	71.63 ± 0.67	71.12 ± 0.63	**76.19** ± 0.62	70.76 ± 0.82	73.57 ± 0.50	73.97 ± 0.59	**77.90** ± 0.51
20	70.79 ± 0.66	74.69 ± 0.42	75.22 ± 0.48	**79.82** ± 0.22	73.74 ± 0.54	75.97 ± 0.37	77.33 ± 0.38	**79.67** ± 0.31
30	75.35 ± 0.47	77.55 ± 0.32	79.06 ± 0.34	**80.42** ± 0.26	75.30 ± 0.43	77.46 ± 0.30	78.16 ± 0.34	**80.77** ± 0.24
40	77.41 ± 0.38	78.93 ± 0.31	80.63 ± 0.33	**80.78** ± 0.25	76.22 ± 0.37	77.27 ± 0.30	78.85 ± 0.29	**81.07** ± 0.24
Ionosphere								
10	72.35 ± 0.66	75.98 ± 0.45	75.26 ± 0.55	**78.56** ± 0.62	74.12 ± 0.71	82.48 ± 0.66	83.23 ± 0.57	**87.73** ± 0.57
20	75.41 ± 0.50	77.10 ± 0.42	77.35 ± 0.34	**82.66** ± 0.34	81.64 ± 0.71	86.75 ± 0.51	87.89 ± 0.44	**91.18** ± 0.51
30	76.97 ± 0.36	77.60 ± 0.37	78.34 ± 0.34	**83.36** ± 0.26	87.02 ± 0.48	89.08 ± 0.37	90.29 ± 0.27	**93.00** ± 0.23
40	77.63 ± 0.30	78.44 ± 0.34	80.08 ± 0.27	**84.52** ± 0.19	89.38 ± 0.36	90.49 ± 0.27	91.07 ± 0.24	**93.07** ± 0.26
Sonar								
10	63.71 ± 0.54	66.84 ± 0.51	68.21 ± 0.49	**68.17** ± 0.62	61.24 ± 0.64	66.28 ± 0.56	64.01 ± 0.57	**69.21** ± 0.60
20	65.23 ± 0.54	65.71 ± 0.45	67.53 ± 0.44	**70.65** ± 0.51	64.72 ± 0.65	67.28 ± 0.59	66.78 ± 0.59	**71.52** ± 0.44
30	66.34 ± 0.52	67.05 ± 0.45	68.16 ± 0.50	**72.77** ± 0.39	68.33 ± 0.55	70.02 ± 0.50	68.89 ± 0.43	**72.79** ± 0.36
40	66.03 ± 0.50	66.49 ± 0.48	68.06 ± 0.48	**73.85** ± 0.36	69.42 ± 0.56	70.70 ± 0.48	70.51 ± 0.50	**73.59** ± 0.34
Wine								
10	81.97 ± 0.79	87.42 ± 0.60	87.44 ± 0.66	**92.37** ± 0.84	88.77 ± 0.63	94.13 ± 0.33	94.47 ± 0.32	**95.63** ± 0.32
20	86.28 ± 0.69	92.66 ± 0.35	93.65 ± 0.40	**96.14** ± 0.19	93.67 ± 0.41	95.29 ± 0.25	96.75 ± 0.18	**97.36** ± 0.13
30	93.08 ± 0.30	94.99 ± 0.23	96.72 ± 0.17	**97.35** ± 0.14	94.34 ± 0.29	95.59 ± 0.21	97.25 ± 0.13	**98.12** ± 0.09
40	95.52 ± 0.24	96.26 ± 0.19	98.06 ± 0.13	**98.39** ± 0.11	96.06 ± 0.14	96.46 ± 0.15	97.96 ± 0.13	**98.34** ± 0.17

we formulate the SKL algorithm into a QP problem, we typically can solve it in quadratic time with respect to the cut-off dimension d, i.e., with a time complexity of $\mathcal{O}(d^2)$. For large problems, d usually is a small number compared with the size of dataset n. If we make a reasonable assumption that $d \leq \sqrt{n}$, then the SKL algorithm can be run in $\mathcal{O}(n)$, i.e., a linear time complexity. In practice, the algorithm could be conducted more efficiently than the linear time complexity with small d and fast implementation of quadratic program.

However, before the SKL algorithm, we need to get the top eigenvectors of the kernel matrix. In general, the time complexity of eigen-decomposition is $\mathcal{O}(n^3)$. Since we only consider top d eigenvectors, we can reduce the time complexity of the eigen-decomposition step to $\mathcal{O}(n^2)$ by adopting some fast decomposition algorithms. Furthermore, if we explore the sparsity of kernel matrix and study some approximation algorithms, such as lower-rank approximation [1], kernel factorization methods [151, 149], sampling-based methods [82, 35], or block-quantized kernel matrix decomposition [161], we can reduce the time complexity to $O(m^2 \times n)$ or $O(m \times n)$, where $m \ll n$. This makes our solution scalable to large-scale problems.

3.7 Summary

This chapter presented a novel general framework of learning the Unified Kernel Machines (UKM) for classification. Different from traditional classification schemes, our UKM framework integrates supervised learning, semi-supervised learning, unsupervised kernel design and active learning in a unified solution, making it more effective for classification tasks. For the proposed framework, we focus our attention on tackling a core problem of learning semi-supervised kernels from labeled and unlabled data. We proposed a Spectral Kernel Learning (SKL) algorithm, which is more effective and efficient for learning kernels from labeled and unlabeled data. Under the framework, we developed a paradigm of unified kernel machine based on Kernel Logistic Regression, i.e., Unified Kernel Logistic Regression (UKLR). Empirical results demonstrated that our proposed solution is more effective than the traditional classification approaches.

□ **End of chapter.**

Chapter 4

Batch Mode Active Learning

4.1 Problem and Motivation

The problem of text categorization is to automatically assign text documents to the predefined categories. One critical issue for large-scale text document categorization is how to reduce the number of labeled documents that are required for building reliable text classification models. Given the limited amount of labeled documents, the key is to exploit the unlabeled documents. One solution is the semi-supervised learning, which tries to learn a classification model from the mixture of labeled and unlabeled examples [131]. A comprehensive study of semi-supervised learning techniques can be found in [118, 170]. Another solution is active learning [90, 119] that tries to choose the most informative unlabeled examples for labeling manually. Although previous studies have shown the promising performance of semi-supervised learning for text categorization [69], the high computation cost has limited its application [170]. In this chapter we focus on active learning for exploiting the unlabeled data for categorization tasks.

In the past, there have been a number of studies on applying active learning to text categorization. The main idea is to only select the most informative documents for labeling manually. Most active learning algorithms are conducted in the iterative fashion. In each iteration, the example with the largest classification uncertainty is chosen for labeling manually. Then, the classification model is retrained with the additional labeled example. The step of training a classification model and the step of soliciting a labeled example are iterated alternatively until most of the examples can be classified with reasonably high confidence.

One of the main problems with most existing active learning algorithm is that only a *single* example is selected for labeling. As a result, the classification model has to be retrained after each labeled example is solicited. In the paper, we propose a novel active learning scheme that is able to select a *batch* of unlabeled examples in each iteration. A simple strategy toward the batch mode active learning is to select the top k most informative examples. The problem with such an approach is that some of the selected examples could be similar, or even identical, and therefore do not provide additional information for model updating.

In general, the key of the batch mode active learning is to ensure the small redundancy among the selected examples such that each example provides unique information for model updating. To this end, we use the Fisher information matrix, which represents the overall uncertainty of a classification model. We choose the set of examples such that the Fisher information of a classification model can be effectively maximized. Fisher information matrix has been used widely in statistics for measuring model uncertainty [123]. For example, in the Cramer-Rao bound, Fisher information matrix provides the low bound for the variance of a statistical model. In this study, we choose the set of examples that can well represent the structure of the Fisher information matrix.

The rest of this chapter is organized as follows. Section 4.2 briefly introduces the concept of logistic regression, which is used as the classification model in our study for text categorization. Section 4.3 presents the batch mode active learning algorithm and an efficient learning algorithm based on the bound optimization algorithm. Section 4.4 presents the results of our empirical study. Section 5.7 summarizes this chapter.

4.2 Logistic Regression

In this section, we give a brief review of logistic regression, which has been a well-known and mature statistical model suitable for probabilistic binary classification. Recently, logistic regression has been actively studied in statistical machine learning community due to its close relation to SVMs and Adaboost [141, 160].Compared with many other statistical learning models, such as SVMs, the logistic regression model has the following advantages:

- It is a high performance classifier that can be efficiently trained with a large number of labeled examples. Previous studies have shown that the logistic regression model is able to achieve the similar performance of text categorization as SVMs [75, 160]. These studies also showed that the logistic regression model can be trained significantly more efficiently than SVMs, particularly when the number of labeled documents is large.

- It is a robust classifier that does not have any configuration parameters to tune. In contrast, some state-of-the-art classifiers, such as support vector machines and AdaBoost, are sensitive to the setup of the configuration parameters. Although this problem can be partially solved by the cross validation method, it usually introduces a significant amount of overhead in computation.

Logistic regression can be applied to both real and binary data. It outputs the posterior probabilities for test examples that can be conveniently processed and engaged in other systems. In theory, given a test example $\mathbf{x}$, logistic regression models the conditional probability of assigning a class label y to the example by

$$p(y|\mathbf{x}) = \frac{1}{1 + \exp(-y\alpha^T\mathbf{x})} \tag{4.1}$$

where $y \in \{+1, -1\}$, and α is the model parameter. Here a bias constant is omitted for simplified notation. In general, logistic regression is a linear classifier that has been shown effective in classifying text documents that are usually in the high-dimensional data space. For the implementation of logistic regressions, a number of efficient algorithms have been developed in the recent literature [75].

4.3 Batch Mode Active Learning

In this section, we present a batch mode active learning algorithm for large-scale text categorization. In our proposed scheme, logistic regression is used as the base classifier for binary classification. In the following, we first introduce the theoretical foundation of our active learning algorithm. Based on the theoretical framework, we then formulate the active learning problem into a semi-definite programming (SDP) problem [19]. Finally, we present an efficient learning algorithm for the related optimization problem based on the eigen space simplification and a bound optimization strategy.

4.3.1 Theoretical Foundation

Our active learning methodology is motivated by the work in [164], in which the author presented a theoretical framework of active learning based on the Fisher information matrix. Given the Fisher information matrix represents the overall uncertainty of a classification model, our goal is to search for a set of examples that can most efficiently maximize the Fisher information. As showed in [164], this goal can be formulated into the following optimization problem:

Let $p(\mathbf{x})$ be the distribution of all unlabeled examples, and $q(\mathbf{x})$ be the distribution of unlabeled examples that are chosen for labeling manually. Let α denote the parameters of the classification model. Let $I_p(\alpha)$ and $I_q(\alpha)$ denote the Fisher information matrix of the classification model for the distribution $p(\mathbf{x})$ and $q(\mathbf{x})$, respectively. Then, the set of examples that can most efficiently reduce the uncertainty of classification model is found by minimizing the ratio of the two Fisher information matrix $I_p(\alpha)$ and $I_q(\alpha)$, i.e.,

$$q^* \;=\; \arg\min_q \mathbf{tr}(I_q(\alpha)^{-1} I_p(\alpha)) \tag{4.2}$$

For the logistic regression model, the Fisher information $I_q(\alpha)$ is attained as:

$$\begin{aligned} & I_q(\alpha) \\ & = -\int q(\mathbf{x}) \sum_{y=\pm 1} p(y|\mathbf{x}) \frac{\partial^2}{\partial \alpha^2} \log p(y|\mathbf{x}) d\mathbf{x} \\ & = \int \frac{1}{1+\exp(\alpha^T \mathbf{x})} \frac{1}{1+\exp(-\alpha^T \mathbf{x})} \mathbf{x}\mathbf{x}^T q(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{4.3}$$

In order to estimate the optimal distribution $q(\mathbf{x})$, we replace the integration in the above equation with the summation over the unlabeled data, and the model parameter α with the empirically estimated $\hat{\alpha}$. Let $D = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be the unlabeled data. We can now rewrite the above expression for Fisher information matrix as:

$$I_q(\hat{\alpha}) = \sum_{i=1}^{n} \pi_i (1 - \pi_i) \mathbf{x}_i \mathbf{x}_i^T q_i + \delta I_d \tag{4.4}$$

where

$$\pi_i = p(-|\mathbf{x}_i) = \frac{1}{1+\exp(\hat{\alpha}^T \mathbf{x}_i)} \tag{4.5}$$

In the above, q_i stands for the probability of selecting the i-th example and is subjected to $\sum_{i=1}^n q_i = 1$, I_d is the identity matrix of d dimension, and δ is the smoothing parameter. The δI_d term is added to the estimation of $I_q(\hat{\alpha})$ to prevent it from being a singular matrix. Similarly, for $I_p(\hat{\alpha})$, the Fisher information matrix for all the unlabeled examples, we have it expressed as follows:

$$I_p(\hat{\alpha}) = \frac{1}{n}\sum_{i=1}^{n} \pi_i(1-\pi_i)\mathbf{x}_i\mathbf{x}_i^T + \delta I_d \tag{4.6}$$

4.3.2 Why Using Fisher Information Matrix?

In this section, we will qualitatively justify the theory of minimizing the Fisher information for batch mode active learning. In particular, we consider two cases, the case of selecting a single unlabeled example and the case of selecting two unlabeled examples simultaneously. To simplify our discussion, let's assume $\|\mathbf{x}_i\|_2^2 = 1$ for all unlabeled examples.

Selecting a single unlabeled example. The Fisher information matrix I_q is simplified into the following form when the i-th example is selected:

$$I_q(\hat{\alpha}; \mathbf{x}_i) \quad = \quad \pi_i(1-\pi_i)\mathbf{x}_i\mathbf{x}_i^T + \delta I_d$$

Then, the objective function $\mathbf{tr}(I_q(\hat{\alpha})^{-1}I_p(\hat{\alpha}))$ becomes:

$$\begin{aligned}&\mathbf{tr}(I_q(\hat{\alpha})^{-1}I_p(\hat{\alpha})) \approx \\ &\quad \frac{1}{n\pi_i(1-\pi_i)}\sum_{j=1}^{n}\pi_j(1-\pi_j)(\mathbf{x}_i^T\mathbf{x}_j)^2 \\ &\quad + \frac{1}{n\delta}\sum_{j=1}^{n}\pi_j(1-\pi_j)(1-(\mathbf{x}_i^T\mathbf{x}_j)^2)\end{aligned}$$

To minimize the above expression, we need to maximize the term $\pi_i(1-\pi_i)$, which reaches its maximum value at $\pi_i = 0.5$. Since $\pi_i = p(-|\mathbf{x}_i)$, the value of $\pi_i(1-\pi_i)$ can be regarded as the measurement of classification uncertainty for the i-th unlabeled example. Thus, the optimal example chosen by minimizing the Fisher information matrix in the above expression tends to be the one with a high classification uncertainty.

Selecting two unlabeled examples simultaneously. To simplify our discussion, we assume that the three examples, $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, have the largest classification uncertainty. Let's further assume that $\mathbf{x}_1 \approx \mathbf{x}_2$, and meanwhile $\mathbf{x}_3$ is far away from $\mathbf{x}_1$ and $\mathbf{x}_2$. Then, if we follow the simple greedy approach, the two example $\mathbf{x}_1$ and $\mathbf{x}_2$ will be selected given their largest classification uncertainty. Apparently, this is not an optimal strategy given both examples provide almost identical information for updating the classification model. Now, if we follow the criterion of minimizing Fisher information, this mistake could be prevented because

$$\begin{aligned} I_q(\hat{\alpha}; \mathbf{x}_1, \mathbf{x}_2) &= \frac{1}{2}(\mathbf{x}_1\mathbf{x}_1^T + \mathbf{x}_2\mathbf{x}_2^T) + \delta I_d \\ &\approx \mathbf{x}_1\mathbf{x}_1^T + \delta I_d = I_q(\hat{\alpha}; \mathbf{x}_1) \end{aligned}$$

As indicated in the above equation, by including the second example $\mathbf{x}_2$, we did not change the expression of I_q, the Fisher information matrix for the selected examples. As a result, there will be no reduction in the objective function $\mathrm{tr}(I_q(\hat{\alpha})^{-1}I_p(\hat{\alpha}))$ when including the example $\mathbf{x}_2$. Instead, we may want to choose $\mathbf{x}_3$ that is more likely to decrease the objective function even though its classification uncertainty is smaller than that of $\mathbf{x}_2$.

4.3.3 Optimization Formulation

The idea of our batch mode active learning approach is to search a distribution $q(x)$ that minimizes $\mathbf{tr}(I_q^{-1}I_p)$. The samples with maximum values of $q(x)$ will then be chosen for queries. However, it is usually not easy to find an appropriate distribution $q(x)$ that minimizes $\mathbf{tr}(I_q^{-1}I_p)$. In the following, we present an SDP approach for optimizing $\mathbf{tr}(I_q^{-1}I_p)$.

Given the optimization problem in (5.4), we can rewrite the objective function $\mathbf{tr}(I_q^{-1}I_p)$ as $\mathbf{tr}(I_p^{1/2}I_q^{-1}I_p^{1/2})$. We then introduce a slack matrix $M \in \mathbf{R}^{n\times n}$ such that $M \succeq I_p^{1/2}I_q^{-1}I_p^{1/2}$. Then original optimization problem can be rewritten as follows:

$$\begin{aligned} \min_{\mathbf{q},M} \quad & \mathbf{tr}(M) \\ \text{s. t.} \quad & M \succeq I_p^{1/2} I_q^{-1} I_p^{1/2} \\ & \sum_{i=1}^{n} q_i = 1, q_i \geq 0, i = 1, \ldots, n \end{aligned} \tag{4.7}$$

In the above, we use the property $\mathbf{tr}(A) \geq \mathbf{tr}(B)$ if $A \succeq B$. Furthermore, we use the Schur complementary, i.e.,

$$D \succeq AB^{-1}A^T \Leftrightarrow \begin{pmatrix} B & A^T \\ A & D \end{pmatrix} \succeq 0 \tag{4.8}$$

if $B \succeq 0$. This will lead to the following formulation of the problem in (4.7)

$$\begin{aligned} \min_{\mathbf{q},M} \quad & \mathbf{tr}(M) \\ \text{s. t.} \quad & \begin{pmatrix} I_q & I_p^{1/2} \\ I_p^{1/2} & M \end{pmatrix} \succeq 0 \\ & \sum_{i=1}^{n} q_i = 1, q_i \geq 0, i = 1, \ldots, n \end{aligned} \tag{4.9}$$

or more specifically

$$\begin{aligned} \min_{\mathbf{q},M} \quad & \mathbf{tr}(M) \\ \text{s. t.} \quad & \begin{pmatrix} \sum_{i=1}^{n} q_i \pi_i (1-\pi_i) \mathbf{x}_i \mathbf{x}_i^T & I_p^{1/2} \\ I_p^{1/2} & M \end{pmatrix} \succeq 0 \\ & \sum_{i=1}^{n} q_i = 1, q_i \geq 0, i = 1, \ldots, n \end{aligned} \tag{4.10}$$

The above problem belongs to the family of Semi-definite programming and can be solved by the standard convex optimization packages such as SeDuMi [128].

4.3.4 Eigen Space Simplification

Although the formulation in (4.10) is mathematically sound, directly solving the optimization problem could be computationally expensive due to the large size of matrix M, i.e., $d \times d$, where d is the dimension of data. In order to reduce the computational complexity, we assume that M is only expanded in the eigen space of matrix I_p. Let $\{(\lambda_1, \mathbf{v}_1), \ldots, (\lambda_s, \mathbf{v}_s)\}$ be the top s eigen vectors of matrix I_p where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_s$. We assume matrix M has the following form:

$$M = \sum_{k=1}^{s} \gamma_k \mathbf{v}_k \mathbf{v}_k^T \tag{4.11}$$

where the combination parameters $\gamma_k \geq 0,\ k = 1, \ldots, s$. We rewrite the inequality for $M \succeq I_p^{1/2} I_q^{-1} I_p^{1/2}$ as $I_q \succeq I_p^{1/2} M^{-1} I_p^{1/2}$. Using the expression for M in (4.11), we have

$$I_p^{1/2} M^{-1} I_p^{1/2} = \sum_{k=1}^{s} \gamma_k^{-1} \lambda_k \mathbf{v}_k \mathbf{v}_k^T \tag{4.12}$$

Given that the necessary condition for $I_q \succeq I_p^{1/2} M^{-1} I_p^{1/2}$ is

$$\mathbf{v}^T I_q \mathbf{v} \geq \mathbf{v}^T I_p^{1/2} M^{-1} I_p^{1/2} \mathbf{v},\ \forall \mathbf{v} \in \mathbf{R}^d\ ,$$

we have $\mathbf{v}_k^T I_q \mathbf{v}_k \geq \gamma_k^{-1} \lambda_k$ for $k = 1, \ldots, s$. This necessary condition leads to following constraints for γ_k:

$$\gamma_k \geq \frac{\lambda_k}{\sum_{i=1}^{n} q_i \pi_i (1 - \pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2},\ k = 1, \ldots, s \tag{4.13}$$

Meanwhile, the objective function in (4.10) can be expressed as

$$\mathbf{tr}(M) = \sum_{k=1}^{s} \gamma_k \tag{4.14}$$

By putting the above two expressions together, we transform the SDP problem in (4.10) into the following optimization problem:

$$\begin{aligned} \min_{\mathbf{q} \in \mathbf{R}^n} & \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^{n} q_i \pi_i (1 - \pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2} \\ \text{s.t.} & \sum_{i=1}^{n} q_i = 1, q_i \geq 0, i = 1, \ldots, n \end{aligned} \tag{4.15}$$

Note that the above optimization problem is a convex optimization problem since $f(x) = 1/x$ is convex when $x \geq 0$. In the next subsection, we present a bound optimization algorithm for solving the optimization problem in (4.15).

4.3.5 Bound Optimization Algorithm

The main idea of bound optimization algorithm is to update the solution iteratively. In each iteration, we will first calculate the difference between the objective function of the current iteration and the objective function of the previous iteration. Then, by minimizing the upper bound of the difference, we find the solution of the current iteration.

Let $\mathbf{q}'$ and $\mathbf{q}$ denote the solutions obtained in two consecutive iterations, and let $\mathcal{L}(\mathbf{q})$ be the objective function in (4.15). Based on the proof given in Appendix A.2, we have the following expression:

$$\begin{aligned}\mathcal{L}(\mathbf{q}) &= \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^{n} q_i \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2} \\ &\leq \sum_{i=1}^{n} \frac{(q_i')^2}{q_i} \pi_i (1-\pi_i) \sum_{k=1}^{s} \frac{(\mathbf{x}_i^T \mathbf{v}_k)^2 \lambda_k}{\left(\sum_{j=1}^{n} q_j' \pi_j (1-\pi_j)(\mathbf{x}_j^T \mathbf{v}_k)^2\right)^2}\end{aligned} \tag{4.16}$$

Now, instead of optimizing the original objective function $\mathcal{L}(\mathbf{q})$, we can optimize its upper bound, which leads to the following simple updating equation:

$$\begin{aligned} q_i &\longleftarrow q_i^2 \pi_i (1-\pi_i) \sum_{k=1}^{s} \frac{(\mathbf{x}_i^T \mathbf{v}_k)^2 \lambda_k}{\left(\sum_{j=1}^{n} q_j \pi_j (1-\pi_j)(\mathbf{x}_j^T \mathbf{v}_k)^2\right)^2} \\ q_i &\longleftarrow \frac{q_i}{\sum_{j=1}^{n} q_j}\end{aligned} \tag{4.17}$$

Similar to all bound optimization algorithms [19], this algorithm will guarantee to converge to a local maximum. Since the original optimization problem in (4.15) is a convex optimization problem, the above updating procedure will guarantee to converge to a global optimal.

Remark: It is interesting to examine the property of the solution obtained by the updating equation in (4.17). First, according to (4.17), the example with a large classification uncertainty will be assigned with a large probability. This is because q_i is proportional to $\pi_i(1 - \pi_i)$, the classification uncertainty of the i-the unlabeled example. Second, according to (4.17), the example that is similar to many unlabeled examples is more likely to be selected. This is because probability q_i is proportional to the term $(\mathbf{x}_i^T \mathbf{v})^2$, the similarity of the i-th example to the principle eigenvectors. This is consistent with our intuition that we should select the most informative and representative examples for active learning.

4.4 Experimental Results

4.4.1 Experimental Testbeds

Category	# of total samples
earn	3964
acq	2369
money-fx	717
grain	582
crude	578
trade	485
interest	478
wheat	283
ship	286
corn	237

Table 4.1: A list of 10 major categories of the Reuters-21578 dataset in our experiments.

In this section we discuss the experimental evaluation of our active learning algorithm in comparison to the state-of-the-art approaches. For a consistent evaluation, we conduct our empirical comparisons on

Category	# of total samples
course	930
department	182
faculty	1124
project	504
staff	137
student	1641

Table 4.2: A list of 6 categories of the WebKB dataset in our experiments.

Category	# of total samples
0	1000
1	1000
2	1000
3	1000
4	1000
5	1000
6	999
7	1000
8	1000
9	1000
10	997

Table 4.3: A list of 11 categories of the Newsgroup dataset in our experiments.

three standard datasets for text document categorization. For all three datasets, the same pre-processing procedure is applied: the stopwords and the numbers are removed from the documents, and all the words are converted into the low cases without stemmming.

The first dataset is the Reuters-21578 Corpus dataset, which has been widely used as a testbed for evaluating algorithms for text categorization. In our experiments, the ModApte split of the Reuters-21578 is used. There are a total of 10788 text documents in this collection. Table 4.1 shows a list of the 10 most frequent categories contained in the dataset. Since each document in the dataset can be assigned to multiple categories, we treat the text categorization problem as a set of binary classification problems, i.e., a different binary classification problem for each category. In total, $26,299$ word features are extracted and used to represent the text documents.

The other two datasets are Web-related: the WebKB data collection and the Newsgroup data collection. The WebKB dataset comprises of the WWW-pages collected from computer science departments of various universities in January 1997 by the World Wide Knowledge Base (Web->Kb) project of the CMU text learning group. All the Web pages are classified into seven categories: student, faculty, staff, department, course, project, and other. In this study, we ignore the category of others due to its unclear definition. In total, there are 4518 data samples in the selected dataset, and $19,686$ word features are extracted to represent the text documents. Table 4.2 shows the details of this dataset. The newsgroup dataset includes $20,000$ messages from 20 different newsgroups. Each newsgroup contains roughly about 1000 messages. In this study, we randomly select 11 out of 20 newsgroups for evaluation. In total, there are $10,996$ data samples in the selected dataset, and $47,410$ word features are extracted to represent the text documents. Table 4.3 shows the details of the engaged dataset.

Compared to the Reuter-21578 dataset, the two Web-related data collections are different in that more unique words are found in the Web-related datasets. For example, for both the Reuter-21578 dataset and the Newsgroup dataset, they both contain roughly $10,000$ documents. But, the number of unique words for the Newgroups dataset is close to $50,000$, which is about twice as the number of unique words found in the Reuter-21578. It is this feature that makes the text categorization of WWW documents more challenging than the categorization of normal text documents because considerably more feature weights need to be decided for the WWW documents than the normal documents. It is also this feature that makes the active learning algorithms more valuable for text categorization of WWW documents than the normal documents since by selecting the informative documents for labeling manually, we are able to decide the appropriate weights for more words than by a randomly chosen document.

4.4.2 Experimental Settings

In order to remove the uninformative word features, feature selection is conducted using the Information Gain [158] criterion. In particular, 500 of the most informative features are selected for each category in each of the three datasets above.

For performance measurement, the $F1$ metric is adopted as our evaluation metric, which has been shown to be more reliable metric than other metrics such as the classification accuracy [158]. More specifically, the $F1$ is defined as

$$F1 = \frac{2 * p * r}{p + r} \tag{4.18}$$

where p and r are *precision* and *recall.* Note that the $F1$ metric takes into account both the precision and the recall, thus is a more comprehensive metric than either the precision or the recall when separately considered.

To examine the effectiveness of the proposed active learning algorithm, two reference models are used in our experiment. The first reference model is the logistic regression active learning algorithm that measures the classification uncertainty based on the entropy of the distribution $p(y|\mathbf{x})$. In particular, for a given test example $\mathbf{x}$ and a logistic regression model with the weight vector $\mathbf{w}$ and the bias term b, the entropy of the distribution $p(y|\mathbf{x})$ is calculated as:

$$H(p) = -p(-|\mathbf{x}) \log p(-|\mathbf{x}) - p(+|\mathbf{x}) \log p(+|\mathbf{x})$$

The larger the entropy of $\mathbf{x}$ is, the more uncertain we are about the class labels of $\mathbf{x}$. We refer to this baseline model as the logistic regression active learning, or **LogReg-AL** for short. The second reference model is based on support vector machine [137]. In this method, the classification uncertainty of an example $\mathbf{x}$ is determined by its distance to the decision boundary $\mathbf{w}^T\mathbf{x} + b = 0$, i.e.,

$$d(\mathbf{x}; \mathbf{w}, b) = \frac{|\mathbf{w}^T\mathbf{x} + b|}{\|\mathbf{w}\|_2}$$

The smaller the distance $d(\mathbf{x}; \mathbf{w}, b)$ is, the more the classification uncertainty will be. We refer to this approach as support vector machine active learning, or **SVM-AL** for short. Finally, both the logistic regression model and the support vector machine that are trained only over the labeled examples are used in our experiments as the baseline models. By comparing with these two baseline models, we are able to determine the amount of benefits that are brought by different active learning algorithms.

To evaluate the performance of the proposed active learning algorithms, we first pick 100 training samples, which include 50 positive examples and 50 negative examples, randomly from the dataset for each category. Both the logistic regression model and the SVM classifier are trained on the labeled data. For the active learning methods, 100 unlabeled data samples are chosen for labeling and their performances are evaluated after rebuilding the classifiers respectively. Each experiment is carried out 40 times and the averaged $F1$ with its variance is calculated and used for final evaluation.

To deploy efficient implementations of our scheme toward large-scale text categorization tasks, all the algorithms used in this study are programmed in the C language. The testing hardware environment is on a Linux workstation with 3.2GHz CPU and $2GB$ physical memory. To implement the logistic regression algorithm for our text categorization tasks, we employ the implementation of the logistic regression tool developed by Komarek and Moore recently [75]. To implement our active learning algorithm based on the bound optimization approach, we employ a standard math package, i.e., LAPACK [2], to solve the eigen decomposition in our algorithm efficiently. The SVM^{light} package [68] is used in our experiments for the implementation of SVM, which has been considered as the state-of-the-art tool for text categorization. Since SVM is not parameter-free and can be very sensitive to the capacity parameter, a separate validation set is used to determine the optimal parameters for configuration.

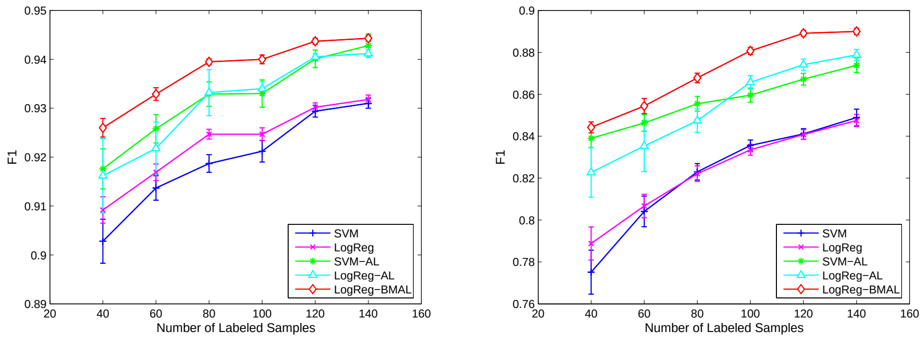
4.4.3 Empirical Evaluation

In this subsection, we will first describe the results for the Reuter-21578 dataset since this dataset has been most extensively studied for text categorization. We will then provide the empirical results for the two Web-related datasets.

Experimental Results with Reuter-21578

Category	SVM	LogReg	SVM-AL	LogReg-AL	LogReg-BMAL
earn	92.12 ± 0.22	92.47 ± 0.13	93.30 ± 0.28	93.40 ± 0.14	**94.00 ± 0.09**
acq	83.56 ± 0.26	83.35 ± 0.26	85.96 ± 0.34	86.57 ± 0.32	**88.07 ± 0.17**
money-fx	64.06 ± 0.60	63.71 ± 0.63	73.32 ± 0.38	71.21 ± 0.61	**75.54 ± 0.26**
grain	60.87 ± 1.04	58.97 ± 0.91	74.95 ± 0.42	74.82 ± 0.53	**77.77 ± 0.27**
crude	67.78 ± 0.39	67.32 ± 0.48	75.72 ± 0.24	74.97 ± 0.44	**78.04 ± 0.14**
trade	52.64 ± 0.46	48.93 ± 0.55	66.41 ± 0.33	66.31 ± 0.33	**69.29 ± 0.34**
interest	56.80 ± 0.60	53.59 ± 0.60	67.20 ± 0.39	66.15 ± 0.49	**68.71 ± 0.37**
wheat	62.71 ± 0.72	57.38 ± 0.79	86.01 ± 1.04	86.49 ± 0.27	**88.15 ± 0.21**
ship	67.11 ± 1.59	64.91 ± 1.75	75.86 ± 0.53	72.82 ± 0.46	**76.82 ± 0.34**
corn	44.39 ± 0.84	41.15 ± 0.69	71.27 ± 0.62	71.61 ± 0.60	**74.35 ± 0.47**

Table 4.4: Experimental results of F1 performance on the Reuters-21578 dataset with 100 training samples (%).



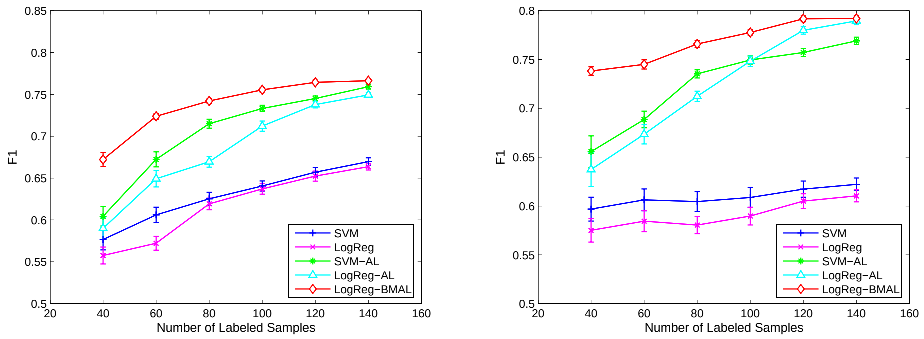
(a) “earn”

(b) “acq”

Figure 4.1: Experimental results of F1 performance on the “earn” and “acq” categories

Table 4.4 shows the experimental results of $F1$ performance averaging over 40 executions on 10 major categories in the dataset.

First, as listed in the first and the second columns of Table 4.4, we observe that the performance of the two classifiers, logistic regression and SVM, are comparable when only the 100 initially labeled examples are used for training. For categories, such as “trade” and ”interest”, SVM achieves noticeably better performance than the logistic regression model. Second, we compare the performance of the two classifiers for active learning, i.e., LogReg-AL and SVM-AL, which are the greedy algorithms and select the most informative examples for labeling manually. The results are listed in the third and the fourth columns of Table 4.4. We find that the performance of these two active learning methods becomes closer than the case when no actively labeled examples are used for training. For example, for category “trade”, SVM performs substantially better than the logistic regression model when only 100 labeled examples are used. The difference in $F1$ measurement between LogReg-AL and SVM-AL almost diminishes when both classifiers use the 100 actively labeled examples for training. Finally, we



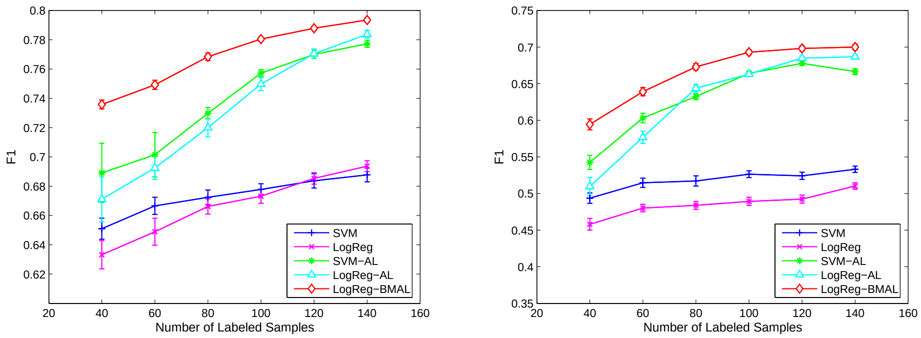
(a) “money-fx”

(b) “grain”

Figure 4.2: Experimental results of F1 performance on the “money-fx” and “grain” categories

compare the performance of the proposed active learning algorithm, i.e., LogReg-BMAL, to the margin-based active learning approaches LogReg-AL and SVM-AL. It is evident that the proposed batch mode active learning algorithm outperforms the margin-based active learning algorithms. For categories, such as “corn” and “wheat”, where the two margin-based active learning algorithms achieve similar performance, the proposed algorithm LogReg-BMAL is able to achieve substantially better $F1$ scores. Even for the categories where the SVM performs substantially better than the logistic regression model, the proposed algorithm is able to outperform the SVM-based active learning algorithm noticeably. For example, for category “ship” where SVM performs noticeably better than the logistic regression, the proposed active learning method is able to achieve even better performance than the margin-based active learning based on the SVM classifier.

In order to evaluate the performance in more detail, we conduct the evaluation on each category by varying the number of initially labeled instances for each classifier. Fig. 4.2, Fig. 4.3 and Fig. 4.4 show the experimental results of the mean $F1$ measurement on 9 major categories.

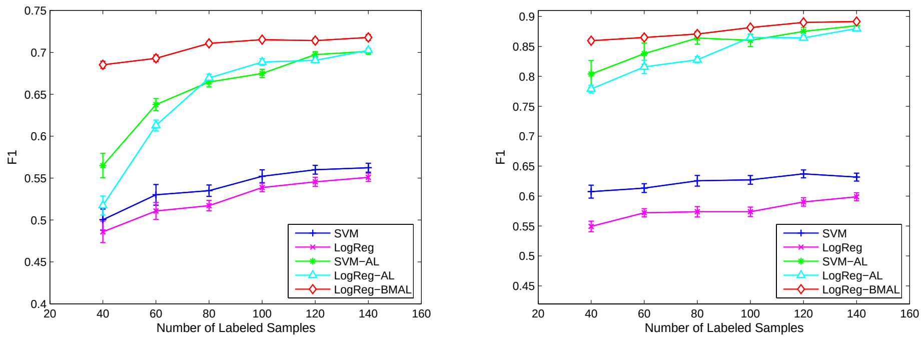


(a) “crude”

(b) “trade”

Figure 4.3: Experimental results of F1 performance on the “crude” and “trade” categories

From the experimental results, we can see that our active learning algorithm outperforms the other two active learning algorithms in most of the cases while the SVM-AL method is generally better than the LogReg-AL method. We also found that the improvement of our active learning method is more evident comparing with the other two approaches when the number of labeled instances is smaller. This is because the smaller the number of initially labeled examples used for training, the larger the improvement we would expect. When more labeled examples are used for training, the gap for future improvement begins to decrease. As a result, the three methods start to behavior similarly. This result also indicates that the proposed active learning algorithm is robust even when the number of labeled examples is small while the other two active learning approaches may suffer critically when the margin criterion is not very accurate for the small sample case.



(a) “interest”

(b) “wheat”

Figure 4.4: Experimental results of F1 performance on the “interest” and “wheat” categories

Category	SVM	LogReg	SVM-AL	LogReg-AL	LogReg-BMAL
course	87.11 ± 0.51	89.16 ± 0.45	88.55 ± 0.48	89.37 ± 0.65	**90.99 ± 0.39**
department	67.45 ± 1.36	68.92 ± 1.39	**82.02 ± 0.47**	79.22 ± 1.14	81.52 ± 0.46
faculty	70.84 ± 0.76	71.50 ± 0.59	75.59 ± 0.65	73.66 ± 1.23	**76.81 ± 0.51**
project	54.06 ± 0.82	56.74 ± 0.57	57.67 ± 0.98	56.90 ± 1.01	**59.71 ± 0.82**
staff	12.73 ± 0.44	12.73 ± 0.28	19.48 ± 1.07	**24.84 ± 0.58**	21.08 ± 0.73
student	74.05 ± 0.51	76.04 ± 0.49	77.03 ± 0.95	80.40 ± 1.16	**81.50 ± 0.44**

Table 4.5: Experimental results of F1 performance on the WebKB dataset with 40 training samples (%).

Experimental Results with Web-Related Datasets

The classification results of the WebKB dataset and the Newsgroup dataset are listed in Table 4.4.3 and Table 4.6, respectively.

First, notice that for the two Web-related datasets, there are a few categories whose $F1$ measurements are extremely low. For example, for the category “staff” of the WebKB dataset, the $F1$ measurement is only about 12% for all methods. This fact indicates that the text categorization of WWW documents can be more difficult than the categorization of normal documents. Second, we observe that the difference in the

Category	SVM	LogReg	SVM-AL	LogReg-AL	LogReg-BMAL
0	96.44 ± 0.35	95.02 ± 0.45	97.37 ± 0.52	95.66 ± 1.01	**98.73 ± 0.11**
1	83.38 ± 1.01	83.12 ± 0.96	**91.61 ± 0.57**	85.07 ± 1.51	91.12 ± 0.36
2	61.03 ± 1.51	59.01 ± 1.39	61.15 ± 2.08	64.91 ± 2.52	**66.13 ± 1.32**
3	72.36 ± 1.90	71.96 ± 1.67	73.15 ± 2.71	75.88 ± 3.13	**78.47 ± 1.95**
4	55.61 ± 1.06	56.09 ±1.21	56.05 ±2.18	61.87 ±2.25	**61.91 ± 1.03**
5	70.58 ± 0.51	72.47 ±0.40	71.69 ±1.11	72.99 ±1.46	**76.54 ± 0.43**
6	85.25 ± 0.45	86.30 ±0.45	89.54 ±1.09	89.14 ±0.89	**92.07 ± 0.26**
7	39.07 ± 0.90	40.22 ±0.90	42.19 ±1.13	46.72 ±1.61	**47.58 ± 0.76**
8	58.67 ± 1.21	59.14 ±1.25	63.77 ±2.05	66.57 ±1.24	**67.07 ± 1.34**
9	69.35 ± 0.82	70.82 ±0.92	74.34 ±1.79	77.17 ±1.06	**77.48 ± 1.20**
10	99.76 ± 0.10	99.40 ±0.21	**99.95 ± 0.02**	99.85 ±0.06	99.90 ± 0.06

Table 4.6: Experimental results of F1 performance on the Newsgroup dataset with 40 training samples (%).

$F1$ measurement between the logistic regression model and the SVM is smaller for both the WebKB dataset and the Newsgroup dataset than for the Reuters-21578 dataset. In fact, there are a few categories in WebKB and Newsgroup that the logistic regression model performs slightly better than the SVM. Third, by comparing the two margin-based approaches for active learning, namely, LogReg-AL and SVM-AL, we observe that, for a number of categories, LogReg-AL achieves substantially better performance than SVM-AL. The most noticeable case is the category 4 of the Newsgroup dataset where the SVM-AL algorithm is unable to improve the $F1$ measurement than the SVM even with the additional labeled examples. In contrast, the LogReg-AL algorithm is able to improve the $F1$ measurement from 56.09% to 61.87%. Finally, comparing the LogReg-BMAL algorithm with the LogReg-AL algorithm, we observe that the proposed algorithm is able to improve the $F1$ measurement substantially over the margin-based approach. For example, for the category 1 of the Newsgroup dataset, the active learning algorithm LogReg-AL only make a slight improvement in the

$F1$ measurement with the additional 100 labeled examples. The improvement for the same category by the proposed batch active learning algorithm is much more significant, increasing from 83.12% to 91.12%. Comparing all the learning algorithms, the proposed learning algorithm achieves the best or close to the best performance for almost all categories. This observation indicates that the proposed active learning algorithm is effective and robust for large-scale text categorization of WWW documents.

4.5 Computational Complexity

We have formulated the framework of batch mode active learning into convex optimization problems. It is interested and important to analyze the algorithms we studied in this work. The first formulation we given in Eqn. 4.10 is an SDP problem. According to convex optimization theory, this problem can be solved in a polynomial time. However, when the dimension of slack matrix M is large, the problem can still be computationally expensive. If we look into the improved solution of the bound optimization algorithm, we can find it can be solved much efficient. When we examine the bound optimization algorithm, we can see that the complexity of the optimization procedure itself is $\mathcal{O}(t \times s \times n)$, where t is the number of iteration steps, s is the number of top eigenvectors, and n is the size of data. Typically, t and s are very small numbers which can be regarded as constants. Therefore, the bound optimization algorithm itself can be done in a linear time complexity. However, before running the bound optimization algorithm, we need to do eigen-decomposition on the Fisher information matrix I_p. In general, the complexity of eigen-decomposition is $\mathcal{O}(n^3)$. But, in our problem, we only need to consider the top eigenvectors. By using fast implementations, we can reduce its complexity to $\mathcal{O}(n^2)$. If we further consider some speedup solution, such as domain decomposition techniques through parallel computing [125], it is possible to further reduce the complexity to $\mathcal{O}(nlogn)$.

4.6 Related Work and Discussions

Active learning has been extensively studied in machine learning [84, 85, 93, 108, 95]. It has been applied to a lot of applications including text classification [85, 137] and content-based image retrieval [136]. Traditional active learning algorithms are often conducted in an iterative fashion. In each iteration, only one example is selected for labeling. Then, the classification model is retrained with the additional labeled example. The active learning theories are typically based on selecting one example with the highest classification uncertainty in each learning iteration. Several different theories have been studied for measuring the classification uncertainties of unlabeled examples. For example, in [95, 39, 40, 44, 76, 93, 119], a number of different classification models are first generated and then the classification uncertainty of a test example is measured by the amount of disagreement among the ensemble of classification models in predicting the test example. Another kind of approaches measure the classification uncertainty of a test example by how far the example is away from the classification boundary [22, 116, 137], such as support vector machine active learning developed by Tong and Koller [137].

The limitation of traditional work in active learning is the lack of theories and effective algorithms for selecting a batch of most informative examples. This is particularly important when studying active learning for large-scale applications. Recently, T. Zhang proposed a theoretical framework of addressing the active learning problem based on Fisher information theory [164]. Our work follows the idea of Fisher information theory and proposed an effective solution for active learning to address how to select a batch of most informative examples for classification tasks. There are also some other existing work to address the sampling strategies of selecting a batch of examples for active learning. For example, K. Brinker proposed to select a batch of examples close to classification boundary and maintain the diversity by measuring the angles between examples [20]. Chang et al. also studied the similar approaches to image retrieval applications [37]. While they considered the sampling problem of selecting a batch of examples, these approaches are usually a bit ad-hoc and lack a theory of how is optimal for a batch selecting solution. Our algorithm is based on the theoretical framework of Fisher information, which is proved to be optimal in a probabilistic framework, though certain reasonable approximation is involved in our fast algorithm.

4.7 Summary

This chapter presented a novel active learning algorithm that is able to select a batch of informative and diverse examples for labeling manually. This is different from traditional active learning algorithms that focus on selecting the most informative examples for manually labeling. We use the Fisher information matrix for the measurement of model uncertainty and choose the set of examples that will effectively maximize the Fisher information matrix. We conducted extensive experimental evaluations on three standard data collections for text categorization. The promising results demonstrate that our method is more effective than the margin-based active learning approaches, which have been the dominating method for active learning. We believe our scheme is essential to performing large-scale categorization of text documents especially for the rapid growth of Web documents on World Wide Web.

□ **End of chapter.**

Chapter 5

Distance Metric Learning

5.1 Problem Definition

In general, the problem of finding a good distance metric for various machine-learning algorithms can be regarded as an equivalent approach of looking for a good data transformation function $f : X \longmapsto Y$, which transforms the data X into another representation of Y [9]. These two problems can be solved together in a unified framework. Hence, the goal is to find a good distance metric which not only can be used for similarity measure of data, but also can transform the data into another better representation of the original data.

Let us introduce some basic concepts for distance metric learning. Mathematically, a linear distance metric learning can be considered as a problem to learn the Mahalanobis metric M, in which the distance between two data instances can be defined as:

$$d_M(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j)} \tag{5.1}$$

where M must be positive semi-definite to satisfy the properties of metric, i.e., non-negativity and triangle inequality. The matrix M can be decomposed as $M = A^\top A$, where A is a transformation matrix. The goal of distance metric learning is to find an optimal Mahalanobis matrix M and the optimal data transformation matrix A based on the contextual information.

For nonlinear distance metric learning, one can map the data into high dimensional space via kernel tricks and learn the corresponding metric in the kernel space. Mathematically, a nonlinear distance metric learning can be mathematically formulated as

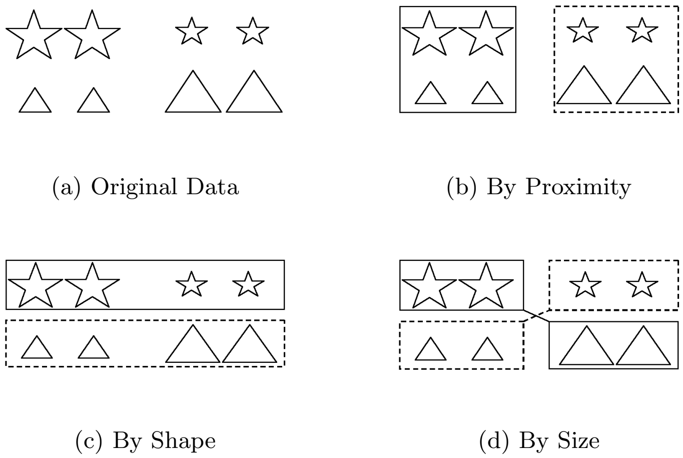
$$d_M(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) = \sqrt{(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j))^\top M(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j))} \qquad (5.2)$$

where ϕ is a basis function that maps the data from the original data space to a higher-dimensional feature space and M the metric defined in the mapping feature space.

5.2 Motivation of Methodology

To learn effective metrics, traditional techniques normally need to acquire explicit class labels. However, in many real-world applications, explicit class labels might be expensive to be obtained. For example, in image retrieval, obtaining the exact class label of images is usually quite expensive due to the difficulty of image annotation. However, it is much easier to know the relevance relationship between images, which can be obtained from the logs of users' relevance feedback [57, 59]. Therefore, it is more attractive to learn the distance metrics or transformation directly from the pairwise constraints without using explicit class labels.

Let us give some example to show that why a flexible distance metric is important for the applications given different kinds of contextual information. Figure 5.1 shows an example of grouping the data instances on different contextual conditions. Figure 5.1 (a) is the given data. Figure 5.1 (b)-(d) show three different grouping results under different context environments, e.g., (b) groups by proximity, (c) groups by shape, (d) groups by size. This example shows that it is important for the clustering algorithms to choose a right distance metric for achieving the correct grouping results under different contextual information.



(a) Original Data

(b) By Proximity

(c) By Shape

(d) By Size

Figure 5.1: Clustering data with different contextual information.

To tackle the problem of learning distance metrics from contextual constraints (also called side-information [153]) among data instances, this thesis first proposes a Discriminative Component Analysis (DCA) method that learn the most discriminative transformation to achieve an optimal linear distance metric. Further, we address the limitation of traditional linear distance metric learning, which may not be effective to capture the nonlinear relationship between data instances in real-world applications. To attack this challenge, a Kernel Discriminative Component Analysis (KDCA) algorithm is then proposed by applying kernel techniques to solve the nonlinear distance metric learning. Note that these methods are different from RCA in which RCA does not consider negative constraints which may provide important discriminative clues in learning metrics. The other difference is that RCA learns only the linear metric while the Kernel DCA method can learn nonlinear distance metrics.

Summary of Contributions. This chapter studies the problem of learning optimal transformations for distance metric learning with contextual constraints in clustering application. Two novel algorithms, namely the DCA and Kernel DCA, are proposed to learn both linear and nonlinear distance metrics. These algorithms need no explicit class labels, which can be applicable to many general applications. The rest of this chapter is organized as follows. Section 5.3 formulates the Discriminative Component Analysis and presents the algorithm. Section 5.4 suggests kernel transformations to extend DCA for learning nonlinear distance metrics. Section 5.5 discusses experimental evaluations on data clustering. Section 5.7 summarizes this chapter.

5.3 Discriminative Component Analysis

5.3.1 Overview

Let us first give an overview of the concept of Discriminative Component Analysis (DCA). In the settings of DCA learning, we assume the data instances are given with contextual constraints which indicate the relevance relationship (positive or negative) between data instances. According to the given constraints, one can group the data instances into chunklets by linking the data instances together with positive constraints. The basic idea of DCA is to learn an optimal data transformation that leads to the optimal distance metric by both maximizing the total variance between the discriminative data chunklets and minimizing the total variance of data instances in the same chunklets. In the following part, we formalize the approach of DCA and present the algorithm to solve the DCA problem.

5.3.2 Formulation

Assume we are given a set of data instances $X = \{\mathbf{x}_i\}_{i=1}^N$ and a set of contextual constraints. Assume that n chunklets can be formed by the positive constraints among the given constraints. For each chunklet, a discriminative set is formed by the negative constraints to represent the discriminative information. For example, for the j-th chunklet, each element in the discriminative set D_j indicates one of n chunklets that can be discriminated from the j-th chunklet. Here, a chunklet is defined to be discriminated from another chunklet if there is at least one negative constraint between them. Note that RCA can be considered as a special case of DCA in which all discriminative sets are empty sets that ignore all negative constraints.

To perform Discriminative Component Analysis, two covariance matrices $\hat{C}_b$ and $\hat{C}_w$ are defined to calculate the total variance between data of the discriminative chunklets and the total variance of data among the same chunklets respectively. These two matrices $\hat{C}_b$ and $\hat{C}_w$ are computed as follows:

$$\begin{aligned} \hat{C}_b &= \frac{1}{n_b}\sum_{j=1}^{n}\sum_{i\in D_j}(\mathbf{m}_j-\mathbf{m}_i)(\mathbf{m}_j-\mathbf{m}_i)^\top \\ \hat{C}_w &= \frac{1}{n}\sum_{j=1}^{n}\frac{1}{n_j}\sum_{i=1}^{n_j}(\mathbf{x}_{ji}-\mathbf{m}_j)(\mathbf{x}_{ji}-\mathbf{m}_j)^\top \end{aligned} \tag{5.3}$$

where $n_b = \sum_{j=1}^{n}|D_j|$, $|\cdot|$ denotes the cardinality of a set, $\mathbf{m}_j$ is the mean vector of the j-th chunklet, i.e., $\mathbf{m}_j = \frac{1}{n_j}\sum_{i=1}^{n_j}\mathbf{x}_{ji}$, $\mathbf{x}_{ji}$ is the i-th data instance in the j-th chunklet, and D_j is the discriminative set in which each element is one of n chunklets that has at least one negative constraint to the j-th chunklet.

The idea of Discriminative Component Analysis is to look for a linear transformation that leads to an optimal distance metric by both maximizing the total variance of data between the discriminative chunklets and minimizing the total variance of data among the same chunklets. The DCA learning task leads to solve the optimization as follows:

$$J(A) = \arg\max_{A}\frac{|A^\top\hat{C}_bA|}{|A^\top\hat{C}_wA|}\,, \tag{5.4}$$

where A denotes the optimal transformation matrix to be learned. When the optimal transformation A is solved, it leads to obtain the optimal Mahalanobis matrix $M = A^\top A$.

5.3.3 Algorithm

According to the Fisher theory [94, 96], the optimal solution in Equation (5.4) is corresponding to the transformation matrix that diagonalizes both the covariance matrices $\hat{C}_b$ and $\hat{C}_w$ simultaneously [86]. To obtain the solution effectively, we propose an algorithm to find the optimal transformation matrix, which was used to solve LDA in the previous study [159]. The details of our algorithm are shown in **Algorithm 1**.

In our algorithm, a matrix U is first found to diagonalize the covariance matrix $\hat{C}_b$ of between-chunklets. After discarding the column vectors with zero eigenvalues, we can obtain a $k*k$ principal sub-matrix D_b of the original diagonal matrix. This procedure leads to obtain a set of projected subspaces, i.e., $Z = RD_b^{-1/2}$, that can best discriminate the chunklets. Further, we form a matrix $C_z = Z^\top \hat{C}_w Z$ and find a matrix V to diagonalize the matrix C_z. If dimension reduction is required, such that r is the desired dimensionality, then we extract the first r column vectors of V with the smallest eigenvalues to form a lower rank matrix $\hat{V}$. This leads to obtain the reduced diagonal matrix $D_w = \hat{V}^\top C_z \hat{V}$. Finally, the optimal transformation matrix and the optimal Mahalanobis Matrix are given as $A = Z\hat{V}D_w^{-1/2}$ and $M = A^\top A$, respectively.

```
Algorithm 1: Discriminative Component Analysis
Input
- a set of $N$ data instances: $X = \{\mathbf{x}_i\}_{i=1}^{N}$
- $n$ chunklets $C_j$ and discriminative sets $D_j$, j=1,...,n
Output
- optimal transformation matrix $A$
- optimal Mahalanobis matrix $M$
Procedure
1. Compute $\hat{C}_b$ and $\hat{C}_w$ by Equation (5.3) ;
2. Diagonalize $\hat{C}_b$ by eigenanalysis
  2.1. Find $U$ to satisfy $U^\top \hat{C}_b U = \Lambda_b$ and $U^\top U = I$, here
       $\Lambda_b$ is a diagonal matrix sorted in increasing order ;
  2.2. Form a matrix $\hat{U}$ by the last $k$ column vectors of $U$
       with nonzero eigenvalues ;
  2.3. Let $D_b = \hat{U}^\top \hat{C}_b \hat{U}$ be the $k * k$ submatrix of $\Lambda_b$ ;
  2.4. Let $Z = \hat{U} D_b^{-1/2}$ and $C_z = Z^\top \hat{C}_w Z$ ;
3. Diagonalize $C_z$ by eigenanalysis
  3.1. Find $V$ to satisfy $V^\top C_z V = \Lambda_w$ and $V^\top V = I$, here
       $\Lambda_w$ is a diagonal matrix sorted in decreasing order ;
  3.2. If dimension reduction is needed, assume the desired
       dimension is $r$, then form $\hat{V}$ by the first $r$ column
       vectors of $V$ with the smallest eigenvalues and let
       $D_w = \hat{V}^\top C_z \hat{V}$ ; otherwise, let $\hat{V} = V$ and $D_w = \Lambda_w$ ;
4. Final Outputs
  $A = Z \hat{V} D_w^{-1/2}$ and $M = A^\top A$ .
End of Algorithm
```

Figure 5.2: The Discriminative Component Analysis Algorithm.

5.4 Kernel Discriminative Component Analysis

5.4.1 Overview

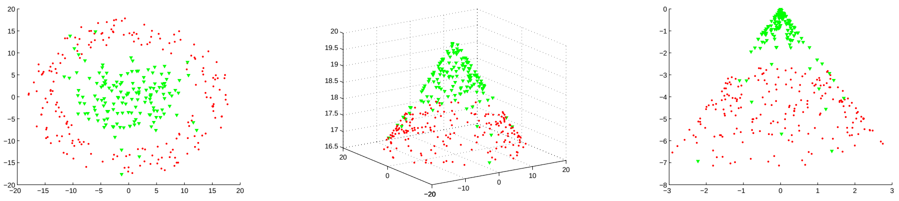
Similar to the RCA learning [9], DCA is so far also a linear technique that is insufficient to discover nonlinear relationships among real-world data. In the machine learning area, the kernel trick is a powerful tool to learn the complex nonlinear structures from the input data [141, 120]. In the literature, the kernel trick has been successfully applied on many linear analysis techniques, such as Kernel Principal Component Analysis (PCA) [148], Kernel Fisher Discriminant Analysis [86, 96], Support Vector Machines [141], Kernel Independent Component Analysis [7], etc. Similar to these approaches, we can also apply the kernel trick on DCA toward more powerful analysis performance in real-world applications.

Although DCA is more advantageous than RCA by incorporating the discriminative information from negative constraints, it is still a linear technique which is inadequate to discover the nonlinear relations among real-world objects. In machine learning, the kernel trick is a powerful tool to represent complicated nonlinear relations of input data. Recently, the kernel-based nonlinear analysis techniques have been successfully applied in a lot of applications, such as Support Vector Machines [13], Kernel Principal Component Analysis (PCA) [14], Kernel FDA [15, 16], Kernel ICA [17], etc.

In general, the kernel technique first maps input data into a high dimensional feature space. A linear technique applied on the data in the feature space is able to achieve the goal of nonlinear analysis. For example, in Kernel PCA, input data are first projected into an implicit feature space via the kernel trick, then the linear PCA is applied on the projected feature space to extract the principal components in the feature space. This enables the Kernel PCA to extract the nonlinear principal components in the input data space using the kernel trick.

Similar to the kernel techniques, we propose the Kernel Discriminative Component Analysis (KDCA) to overcome the disadvantage of RCA and DCA by applying the kernel trick. We first project input data into an implicit feature space via the kernel trick. Then the linear DCA is applied on the projected feature space to find the optimal linear transformation in the feature space. Consequently, we are able to find the nonlinear structures of the given data using the Kernel DCA technique.

To illustrate the capability of capturing nonlinear relations via the kernel trick, we provide a toy example to show different spaces of the given data by using the kernel trick in Figure 5.3. Figure 5.3 (a) shows the original input space of the given data; Figure 5.3 (b) shows the projected space via the kernel trick; Figure 5.3 (c) shows the embedded low-dimensional space via Kernel DCA which be learned by linear DCA. This example demonstrates that the Kernel DCA is more effective to describe the nonlinear relations of the given data.



(a) Original Space

(b) Projected Space

(c) Embedding Space

Figure 5.3: Illustration of Different Data Spaces. (a) is the original data space; (b) is the projected space via kernel tricks; (c) is the embedding space by Kernel DCA learning.

5.4.2 Formulation

Let us now formulate Kernel Discriminative Component Analysis formally. Typically, a kernel-based analysis technique usually implicitly maps original data in input space I to a high-dimensional feature space F via some basis function $\phi : \mathbf{x} \rightarrow \phi(\mathbf{x}) \in F$. The similarity measure of data in the projected feature space is achieved by the kernel function which is defined as an inner product between two vectors in the projected space F as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)). \tag{5.5}$$

Assume that a set of N data instances $X = \{\mathbf{x}_i\}_{i=1}^N$ is given in an original input space I. To do kernel DCA learning, we first choose a basis function ϕ to map the data in the original input space I to a high-dimensional feature space F. For any two data instances, we compute their distance via the kernel function defined in the projected feature space as follows:

$$d_\phi(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j))^\top M (\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j))} \tag{5.6}$$

where M is a full rank matrix that must be positive semi-definite to satisfy the metric property and is often formed by a transformation matrix W. The linear transformation matrix W can be represented as $W = [\mathbf{w}_1, \ldots, \mathbf{w}_m]^\top$ in which each of the m column vectors is a span of all l training samples in the feature space, such that

$$\mathbf{w}_i = \sum_{j=1}^{l} \alpha_{ij} \phi_j \ , \tag{5.7}$$

where α_{ij} are the coefficients to be learned in the feature space. Therefore, for a given data instance $\mathbf{x}$, its projection onto the i-th direction $\mathbf{w}_i$ in the feature space can be computed as follows:

$$(\mathbf{w}_i \cdot \phi(\mathbf{x})) = \sum_{j=1}^{l} \alpha_{ij} K(\mathbf{x}_j, \mathbf{x}) \ . \tag{5.8}$$

Hence, Equation (5.6) can be represented as

$$d_\phi(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\vec{\tau}_i - \vec{\tau}_j)^\top M (\vec{\tau}_i - \vec{\tau}_j)}\ , \tag{5.9}$$

where $\vec{\tau}_i = [K(\mathbf{x}_1, \mathbf{x}_i), \ldots, K(\mathbf{x}_1, \mathbf{x}_l)]^\top$, and A is the linear transformation matrix formed by $A = [\vec{\alpha}_1, \ldots, \vec{\alpha}_m]$ in which $\vec{\alpha}_i = [\alpha_{i1}, \ldots, \alpha_{il}]^\top$. Hence, we can similarly compute the two covariance matrices in the projected feature space as follows:

$$\begin{aligned} K_b &= \frac{1}{n_b} \sum_{j=1}^{n} \sum_{i \in D_j} (\vec{u}_j - \vec{u}_i)(\vec{u}_j - \vec{u}_i)^\top \\ K_w &= \frac{1}{n} \sum_{j=1}^{n} \frac{1}{n_j} \sum_{i=1}^{n_j} (\vec{\tau}_j - \vec{u}_i)(\vec{\tau}_j - \vec{u}_i)^\top \end{aligned} \tag{5.10}$$

where $\vec{u}_j = [\frac{1}{n_j} \sum_{i=1}^{n_j} K(\mathbf{x}_1, \mathbf{x}_i), \ldots, \frac{1}{n_j} \sum_{i=1}^{n_j} K(\mathbf{x}_l, \mathbf{x}_i)]^\top$ denotes the mean vector. Consequently, the Kernel DCA task leads to solve the optimization problem as follows:

$$J(A) = \arg\max_{A} \frac{|A^\top K_b A|}{|A^\top K_w A|}\ . \tag{5.11}$$

Solving the above optimization problem gives the optimal linear transformation A in the projected space. It also leads to the optimal Mahalanobis matrix in the projected space.

5.4.3 Algorithm

The method to solve the optimization of Kernel DCA is similar to that for the linear DCA, i.e., to find the linear transformation matrix A that can diagonalize both K_b and K_w. For limited space, please kindly refer to **Algorithm 2** for the details of Kernel DCA algorithm.

5.5 Experimental Results

To evaluate the performance of our algorithms, we conduct empirical evaluation of learning distance metrics for data clustering in compar-

```
Algorithm 2: Kernel Discriminative Component Analysis
Input
- a set of $N$ data instances: $X = \{\mathbf{x}_i\}_{i=1}^{N}$
- $n$ chunklets $C_j$ and discriminative sets $D_j$, j=1,...,n
Output
- optimal transformation matrix $A$
- optimal Mahalanobis matrix $M$
Procedure
1. Compute $K_b$ and $K_w$ by Equation (5.10) ;
2. Diagonalize $K_b$ by eigenanalysis
  2.1. Find $U$ to satisfy $U^\top K_b U = \Lambda_b$ and $U^\top U = I$, here
       $\Lambda_b$ is a diagonal matrix sorted in increasing order ;
  2.2. Form a matrix $\hat{U}$ by the last $k$ column vectors of $U$
       with nonzero eigenvalues ;
  2.3. Let $D_b = \hat{U}^\top K_b \hat{U}$ be the $k * k$ submatrix of $\Lambda_b$ ;
  2.4. Let $Z = \hat{U} D_b^{-1/2}$ and $K_z = Z^\top K_w Z$ ;
3. Diagonalize $K_z$ by eigenanalysis
  3.1. Find $V$ to satisfy $V^\top K_w V = \Lambda_w$ and $V K_z V = I$, here
       $\Lambda_b$ is a diagonal matrix sorted in decreasing order ;
  3.2. If dimension reduction is needed, assume the desired
       dimension is $r$, then form $\hat{V}$ by the first $r$ column
       vectors of $V$ with the smallest eigenvalues and let
       $D_w = \hat{V}^\top K_z \hat{V}$ ; otherwise, let $\hat{V} = V$ and $D_w = \Lambda_w$ ;
4. Final Outputs
  $A = Z \hat{V} D_w^{-1/2}$ and $M = A^\top A$ .
End of Algorithm
```

Figure 5.4: The Kernel Discriminative Component Analysis Algorithm.

Table 5.1: The six datasets used in our experiments. The first three sets are artificial data and the others are datasets from UCI machine learning repository.

Dataset	**#Classes**	**#Instances**	**#Features**
Norm	2	100	2
Chessboard	2	100	2
Double-Spiral	2	100	3
Iris	3	150	4
Sonar	2	208	60
Wine	3	178	12

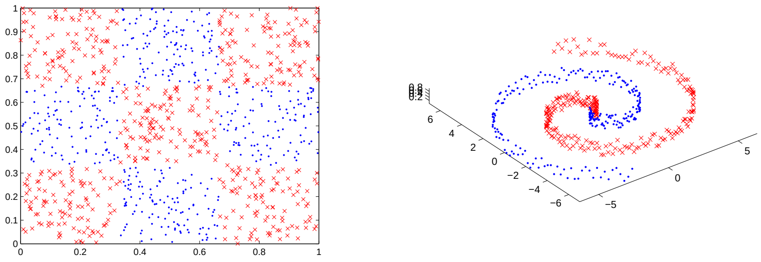
isons with traditional methods of distance metric learning [153, 9]. We describe the details of our empirical evaluation below.

5.5.1 Experimental Testbed

To do the performance evaluation for clustering, we use three artificial datasets and three standard benchmark datasets from UCI machine-learning repository. Table 5.1 shows the details of the datasets employed in our experiment. The Norm artificial dataset was generated by Gaussian distributions. The first feature of the first class was generated by using the Gaussian distribution $N(3, 1)$, and the first feature of the second class by $N(-3, 1)$; the other ten features were generated by $N(0, 25)$. Each feature was generated independently. The other two artificial datasets Chessboard and Double-Spiral are sampled respectively from the data shown in Fig. 5.5.

5.5.2 Performance Evaluation on Clustering

In this experiment, we evaluate the performance of proposed DCA and Kernel DCA in learning distance metrics with contextual constraints for data clustering. Three different techniques are evaluated,



(a) Chessboard Data

(b) Double-Spiral Data

Figure 5.5: Two artificial datasets used in the experiments. 100 data instances are randomly sampled from each to form our datasets respectively.

i.e., RCA [9], Xing's metric learning method [153], DCA, and KDCA. In the experiment, seven different clustering methods are developed for comparison:

(1) K-means-EU: the baseline method, i.e., typical k-means clustering based on the original Euclidean distance;

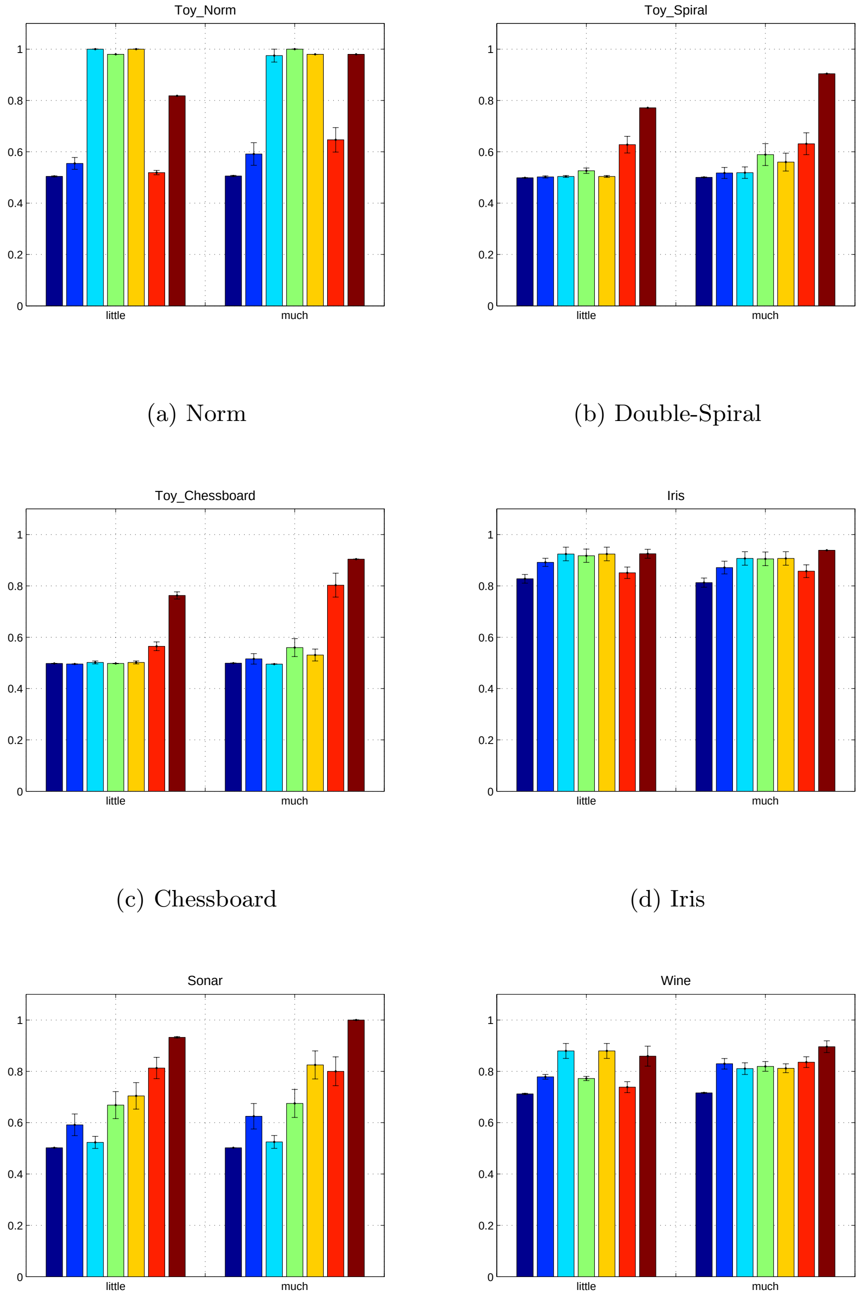
(2) CK-means-EU: the constrained k-means clustering method based on the original Euclidean distance [145];

(3) CKmeans-RCA: the constrained k-means clustering method based on the distance metrics learned by RCA [9];

(4) CKmeans-Xing: the constrained k-means clustering method based on the distance metrics learned by Xing et al. [153];

(5) CKmeans-DCA: the constrained k-means clustering method based on the distance metrics learned by our DCA algorithm;

(6) CKmeans-RBF: the constrained k-means clustering method based on the RBF kernel metrics;



(a) Norm

(b) Double-Spiral

(c) Chessboard

(d) Iris

(e) Sonar

(f) Wine

Figure 5.6: The clustering results on six datasets of several different clustering schemes with different metrics.

(7) CKmeans-KDCA: the constrained k-means clustering method based on the distance metric learned by our Kernel DCA algorithm. The kernel function used in our experiment is a standard RBF function.

In the experiments, we evaluate the performance of the above different clustering schemes given the pairwise contextual constraints. The contextual information is generated automatically based on the ground truth. For any two given data instances in a dataset, if they belong to the same cluster according to the ground truth, this pair is considered as a positive constraint; otherwise, it is a negative constraint. In our experiments, we study two sets of experimental evaluations. One is given with "little" side-information with 5% randomly sampled pairwise constraints (half for similar constraints and half for dissimilar constraints). The other is given with "much" side-information with 10% constraints.

We adopt similar experimental settings for clustering evaluations studied in the previous work [153]. To measure the quality of the clustering results, we adopt the clustering accuracy similar to previous study [9]. The clustering accuracy is defined as follows:

$$Accuracy = \sum_{i>j} \frac{\mathbf{1}\{\mathbf{1}\{c_i = c_j\} = \mathbf{1}\{\hat{c}_i = \hat{c}_j\}\}}{0.5n(n-1)} \tag{5.12}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function ($\mathbf{1}\{true\} = 1$ and $\mathbf{1}\{true\} = 0$), $\{c_i\}_{i=1}^n$ denotes the true cluster of the data instances in the ground truth, $\{\hat{c}_i\}_{i=1}^n$ denotes the cluster predicted by some clustering algorithm, and n is the number of instances in the dataset.

In our experiment, we run 20 times of clustering for each dataset and test their average clustering accuracy performance of the clustering results. Fig. 5.6 shows the experimental results on six datasets using several clustering schemes of different metrics. In each subfigure, the six bars on the left side correspond to an experiment with "little" constraints with 5% pairwise constraints (half for similar pairs and half dissimilar pairs); the bars on the right correspond to “much” constraints with 10% pairwise constraints (half for similar pairs and half for dissimilar pairs). From left to right, the seven bars are respectively K-means+EU, Constrained K-means+EU, Constrained K-means + RCA metric, Constrained K-means + Xing’s metric, Constrained K-means + DCA metric, Constrained K-means + RBF kernel metric, Constrained K-means + Kernel DCA kernel metric. Standard error bars are also shown in the figures.

We conduct the performance analysis on the experimental results. By comparing the linear distance metrics, we can see that CK-means-DCA algorithm perform similar to other two methods, i.e., RCA and Xing’s methods. In some datasets, DCA outperforms the other two methods quite importantly, such as $Sonar$ and $Wine$ datasets. This shows that incorporating the negative constraints by DCA can improve the performance of regular RCA method and the high cost metric learning method by Xing et al. By comparing with the linear and nonlinear distance metric learning using Kernel DCA, we can found that in the $Norm$ dataset where the data relationship is linear, there is not a big difference between linear and nonlinear distance metrics. However, on the $Chessboard$ and $Double-Spiral$ datasets, where the relationship of data instances is nonlinear, the Kernel DCA algorithm significantly outperforms the traditional methods. This shows that Kernel DCA can be more effective and promising in learning nonlinear metric. By examining on the other three benchmark datasets, Kernel DCA algorithms outperform other methods in most cases. This shows that Kernel DCA is more promising in learning metrics compared with traditional metric learning methods.

5.6 Discussions and Future Work

We presented two algorithms, i.e., DCA and Kernel DCA, to overcome the disadvantages of the RCA method. DCA and Kernel DCA also enjoy the merits of simple implementation and efficient computation cost similar to RCA [9]. From the experimental sections, we have empirically demonstrated our methods are promising to learn effective distance functions for the data clustering applications. In addition to learn distance functions, our techniques are also a promising solution for dealing with dimension reduction problems in the appearance of contextual information. We will evaluate the performance of our algorithms on dimension reduction issues in our future work.

Moreover, several problems are also worth studying in our future work. The first problem is the active selection of contextual information. In the setting of our problems, we assume the contextual information could be available for our learning tasks. In some real-world problem, the contextual information may not be adequate. In this case, we can use active learning to solicit users for informative contextual constraints. This is an interesting active learning problem for learning distance functions. One simple and intuitive solution for this problem is to select the constraint between two nearest chunklets in which they have no negative constraints. We can study the problem in more details in our future work.

5.7 Summary

This chapter studied the problem of learning distance metrics and data transformation using the contextual information for data clustering. Two important limitations of traditional RCA approach were addressed. One is the lack of exploiting negative constraints. Another is the limitation of learning the linear distance metrics which are not adequate for describing the complex nonlinear relations of real-world objects. To address the first problem, the Discriminative Component Analysis (DCA) is proposed, which can exploit both positive and negative constraints in an efficient learning scheme. For solving the second problem, the Kernel DCA is proposed by applying the kernel trick on the linear DCA. Empirical evaluations are conducted to evaluate the performance of the algorithms on data clustering. The promising results show that the proposed algorithms are simple but quite effective in learning good quality metrics for data clustering. The methodology studied in this chapter can also be applied to solve other applications, such as web search and multimedia retrieval.

□ **End of chapter.**

Chapter 6

Marginalized Kernels for Web Mining

6.1 Motivation

Although click-through data has received considerable attention on measuring similarity between query terms in the Web research community, most of existing work ignored an important fact, i.e., the similarity between query terms often evolves over time. Here the *similarities* between query terms are usually obtained by the similarity propagation between queries, pages, and their co-occurrences [156]. The dynamic nature of query terms similarity over time can be attributed to many factors, such as seasons, holidays, and special events, etc. This dynamic nature of query terms similarity is usually embedded in the click-through data implicitly. Traditional methods without temporal consideration have limitations to measure such underlying semantic similarity of query terms accurately.

Therefore, it becomes a challenging and important task to develop an effective model for similarity measure from the click-through data that can take advantage of the dynamic nature of queries and pages over time. In particular, two challenging issues are needed to be solved as follows:

- How to exploit the click-through data for semantic similarity measure of queries in terms of temporal consideration?
- How to develop an effective model that can reflect both explicit content similarity and implicit semantics between queries?

To address the challenging issues, let us first illustrate an example to compare two different approaches of measuring similarity between two queries over time in Figure 6.1. In the figure, the dotted line represents the first method that measures the similarity value at each time point based on all the click-through data available at that time. We refer to this method as the *incremented* approach. Different from the *incremented* approach, the other approach, as shown in the solid line, measures the similarity only by the click-through data given at that time interval. We refer to this method as the *interval-based* approach.

From the comparison, we see that the *interval-based* approach can better reflect the temporal factor than the *incremented* approach. For instance, in the *interval-based* approach, the similarity values of the second and third month are as high as *0.8*, while the similarity values in the fourth and fifth month are as low as *0.2*. But in the *incremented* approach, the similarity values from the second month to the fifth month are respectively *0.6, 0.68, 0.55,* and *0.48.* This shows that the similarity values in the *incremented* approach are relatively fixed, which usually cannot evidently reflect the dynamic nature of similarities between the query terms.

Hence, it is important to develop a similarity model that exploits the click-through data effectively in a *time-dependent* way. To this end, we propose a novel time-dependent framework to exploit the click-through data for semantic similarity measure between queries [165]. An overview of our proposed framework is given in the following.

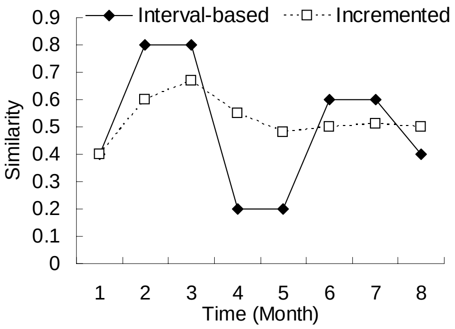


Figure 6.1: Incremented and interval-based methods for similarity measure

6.2 Overview

In this chapter we suggest a time-dependent framework to measure the semantic similarity between Web search queries from the click-through data. More specifically, we propose a novel *time-dependent query term semantic similarity model*, which can exploit the click-through data more effectively than traditional approaches such as the incremented approach shown in Figure 6.1. Our time-dependent model monitors the terms' similarity over the temporal dimension and attempts to discover the *evolution pattern* with the click-through data. Note that in this chapter, the term *evolution pattern* of the query terms' similarity refers to how the similarity values vary over time.

The basic idea of our solution is to construct a model from the click-through data by partitioning the click-through data into sequences of sub-groups with respect to certain predefined *calendar schema* and *calendar patterns*. For example, to monitor query similarity that may change on a daily basis, the click-through data is partitioned into sequences of subgroups, where each sequence consists of click-through data of an individual day. Then, using proposed semantic similarity measure methodology based on the *marginalized kernel function*, a daily based temporal similarity model can be constructed. As a result, the time-dependent model can accurately reflect the daily based patterns such as large or small values of query similarities during the specific days.

Our proposed model can be used for more accurate and efficient query expansion for Web search engines. Our empirical results with the real-world click-through data collected from a commercial search engine show that our proposed model can model the evolution of query terms similarity accurately. In summary, the contributions of our work in this chapter can be summarized as follows:

- To the best of our knowledge, we proposed the first time-dependent model to calculate the query terms similarity by exploiting the dynamic nature of click-through data.
- A probabilistic framework for constructing the time-dependent query term similarity model is proposed with the *marginalized kernel*, which measures both explicit content similarity and implicit semantics from the click-through data.
- Extensive experiments have been done to evaluate the proposed similarity model using a large collection of click-through data collected from a commercial search engine.

The rest of the paper is organized as follows. In Section 2, we review related work of calculating the query terms similarity from the click-through data and temporal analysis of the click-through data. Section 3 presents our time-dependent framework for semantic similarity measure of query terms in the context of query expansion, in which the time-dependent model is formulated by a marginalized kernel. Section 4 provides our empirical evaluation, in which some empirical examples are shown from real-world click-through data and extensive experiments are conducted to evaluate the performance of our model.

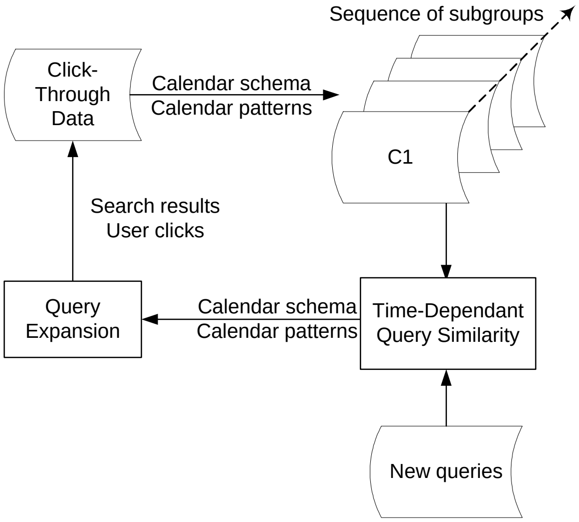


Figure 6.2: A framework of time-dependent semantic similarity measure model between queries

Section 5 discusses limitations and future work. Section 6 concludes this work.

6.3 The Time-Dependent Framework

In this section, we propose a probabilistic framework to construct the time-dependent query similarity model. By exploiting the click-through data over time, a semantic similarity measure model is proposed based on the *marginalized kernel* technique [138]. Figure 6.2 illustrates the framework of our proposed semantic similarity model. The details of our framework will be discussed as the following. First, some preliminary definitions are given to describe click-through data and calendar pattern formally. Following these definitions, a probabilistic approach is proposed to measure content similarity between query terms. Then, we study an efficient clustering technique on the click-through data to acquire the implicit cluster semantics of the data. Lastly, the time-dependent semantic similarity model is formulated by

IP address	Query	Page	Time
xxx.xxx.xxx	BMW	http://www.bmw.com	14:02 02112005
xxx.xxx.xxx	SVM	http://svm.first.gmd.de	15:31 03252005
xxx.xxx.xxx	MSN	http://www.MSN.com	21:14 02142005

Table 6.1: Transaction representation of click-through data.

the marginalized kernel function, which measures both explicit content similarity and implicit cluster semantics effectively from the click-through data.

6.3.1 Click-Through Data & Calendar Pattern

Click-through data, representing query log of Web search engines, keep the records of interactions between Web users and the searching engines. Similar to the transaction data in the supermarket, the click-through data consist of a sequence of users sessions in some format like <*IP address, query, clicked page, time*>. In the literature, there are two different ways to represent the click-through data. One way is to represent the click-through data as session databases where each session represents a pair of a query and a page that the user issued and clicked (hereafter, we call such pairs as *query-page pairs*) as shown in Table 6.1 [32, 147]. Note that the IP addresses are not shown in the table due to the privacy issue. Recently, some variants of this representation have been proposed by taking into account the rank of the page and the ID of the corresponding users [70, 130]. The other representation method is to use a bipartite graph, like the example in Figure 6.3, where the queries and pages are represented as two sets of nodes and the query-page pair co-occurrence relationships are represented as the edges between the corresponding nodes [10, 156].

We use the session database approach to represent the click-through data. Each session is represented as a triple $< \vec{q}, p, \vec{t} >$, where $\vec{q}$ represents the query being issued, p represents the pages being clicked, and

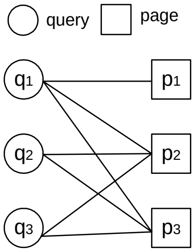


Figure 6.3: Bipartite graph representation of click-through data.

$\vec{t}$ represents the timestamp. Note that we use the vector representation for both the queries and timestamps. The reason is that, in this chapter, the queries will be represented as a vector of related web pages that are led to by the queries. For the timestamps, with a fix *calendar schema*, they are represented as vectors as well. The key distinguishing feature of our query similarity measure is that rather than only using the query-page pairs, the corresponding timestamps are used as well. From the real query log data obtained from a commercial Web search engine, we observe that the granularity of the timestamps can be in mini-second. However, to analyze the temporal patterns of queries, the time granularity can be application dependent and query dependent. To make our framework flexible, we allow users to specify any types of *calendar-based patterns* they are interested in depending on their domain knowledge and application requirements. Formally, the *calendar schema* and *calendar-based pattern* are defined as follows:

Definition 10 Calendar Schema: *A calendar schema, $S = (R, C)$, is a relational schema R with a constraint C, where $R = (f_n : D_n, f_{n-1} : D_{n-1}, \cdots, f_1 : D_1)$, C is a Boolean valid constraint on $D_n \times D_{n-1} \times \cdots \times D_1$ that specifies which combinations of the values in $D_n \times D_{n-1} \times \cdots \times D_1$ are valid.* □

Here each attribute f_i in the relation schema is a calendar unit name such as year, month, week, day, hour, etc. Each domain D_i is a finite subset of positive integers. C is a Boolean function specifying which combinations of the values in $D_n \times D_{n-1} \times \cdots \times D_1$ are valid. For example, we may have calendar schema *(year: $\{$2000, 2001, 2002$\}$, month: $\{$1, 2, 3, $\cdots$,12$\}$, day: $\{$1, 2, 3, $\cdots$, 31$\}$)* with the constraint that evaluate $< y, m, d >$ to be "true" only if the combination gives a valid date. For instance, $<$ *2000, 2, 15* $>$ is valid while $<$ *2000, 2, 30* $>$ is invalid. The reason to use the calendar schema here is to exclude invalid time interval due to the combinations of calendar units. Moreover, by modifying the constraint, users can further narrow down valid time intervals for application and query dependent reasons. Hereafter, we use $*$ to represent any integer value that is valid based on the constraint. For instance, if we use $*$ to represent months in with the above mentioned calendar schema, then $*$ refers to any integer value from *1* to *12*.

Definition 11 Calendar Pattern: *Given a calendar schema $S = (R, C)$, a calendar pattern, denoted as CAP, is a tuple on R of the form $< d_n, d_{n-1}, \cdots, d_1 >$ where $d_i \in D_i \cup \{*\}$.* □

For example, assume we are given the calendar schema $< year, month, day >$, then the calendar pattern $< *, 1, 1 >$ refers to the time intervals "the first day of the first month of every year". Similarly, $< 2002, *, 1 >$ represents the time intervals "the first day of every month in year 2002."

6.3.2 Probabilistic Similarity Measure

Different from the previous query expansion approaches that use the query log and the actual Web pages to extract similarity between terms in the query space and terms in the document space [32, 121], we only employ the query log. We propose a dual approach of the existing information retrieval model by representing each query term as a vector of documents, namely $\vec{q} =< w_1, w_2, \cdots, w_n >$ in which w_i represents the projection weight on the i^{th} page. In our paper, this weight is calculated from the *Page Frequency (PF)* and *Inverted Query Frequency (IQF)*, which are formally defined as follows:

Definition 12 PF.IQF: *Given a query $\vec{q}$ and a Web page p_i that has been clicked by users who issued $\vec{q}$ via the search engine. Then, the Page Frequency (PF) and the Inverted Query Frequency (IQF) are defined as:*

$$PF(\vec{q}, p_i) = \frac{f(\vec{q}, p_i)}{\sum_j f(\vec{q}, p_j)},\ IQF(p_i) = \log \frac{|\vec{q}|}{| < \vec{q}, p_i > |}$$

□

Here $f(\vec{q}, p_i)$ is the number of times that page p_i has been clicked by users who issued the query $\vec{q}$. $\sum_j f(\vec{q}, p_j)$ refers to the total number of times that the pages have been clicked by users who issued the query $\vec{q}$. $|\vec{q}|$ refers to the total number of times that the query $\vec{q}$ has been issued and $| < \vec{q}, p_i > |$ refers to the times that the page p_i has been clicked by users who issued $\vec{q}$. As a result, the weight of page p_i is calculated as $w_i = PF(\vec{q}, p_i) \times IQF(p_i)$.

Based on the document weight vector representation, the similarity between two queries in content can be defined by a cosine kernel function as follows.

Definition 13 Content Similarity Measure: *Given two queries $\vec{q}_1$ and $\vec{q}_2$, their probabilistic similarity in content, denoted as $K_{cos}(\vec{q}_1, \vec{q}_2)$, is defined as:*

$$K_{\cos}(\vec{q}_1, \vec{q}_2) = \frac{\vec{q}_1^T \vec{q}_2}{||\vec{q}_1|| \cdot ||\vec{q}_2||}$$

□

Note that in the above similarity measure, all occurrences of queries and Web pages are considered to be equally important and the timestamps are not used.

6.3.3 Time-Dependent Model via Marginalized Kernels

Based on the content similarity measure and the calendar patterns proposed in the previous section, we now present the framework of the time-dependent query semantic similarity model. Before going into the details of our framework, let us first define the relationship between the timestamp and the calendar pattern to facilitate our following discussions as follows:

Definition 14 Contained: *Given a calendar pattern* $< d_n, d_{n-1}, \cdots, d_1 >$ *denoted as* CAP_i *with the corresponding calendar schema* $S = (R, C)$*. A timestamp* $\vec{t}$ *is represented as* $< d'_n, d'_{n-1}, \cdots, d'_1 >$ *according to* R*.* $\vec{t}$ *is* ***contained*** *in* CAP_i*, denoted as* $\vec{t} \prec CAP_i$*, if and only if* $\forall\ 1 \leq l \leq n$*,* $d'_l \in d_l$*.* □

For example, given a calendar pattern $< *, 2, 12 >$ with the calendar schema $< week, day\ of\ the\ week, hour >$, then the timestamp *2005-09-30 12:28* is not contained in this calendar pattern as it is not the second day of the week, while the timestamp *2005-09-26 12:08* is.

Definition 15 Click-Through Subgroup (CTS): *Given a calendar schema* S *and a set of calendar patterns* $\{CAP_1, CAP_2, \cdots, CAP_m\}$*, the click-through data can be segmented into a sequence of click-through subgroups (CTSs)* $< CTS_1, CTS_2, \cdots, CTS_m >$*, where all query-page pairs* $< \vec{q}, p_i, \vec{t_i} > \in CTS_l$*,* $\vec{t_i} \prec CAP_l$*,* $1 \leq l \leq m$*.* □

With the above definitions, in general, given a collection of click-through data, we can first partition the data into sequences of CTSs based on the user-defined calendar pattern and the corresponding timestamps. For example, given a weekly based calendar schema $< week, day >$ and a list of calendar patterns $< *, 1 >$, $< *, 2 >$, $\cdots$, $< *, 7 >$, the click-through data will be partitioned into sequences of 7 CTSs

$< CTS_1,\ CTS_2,\ \cdots,\ CTS_7 >$, where CTS_i represents the group of click-through data whose timestamps are contained in the i^{th} day of the week.

After that, the query similarities are computed within each subgroup and are aligned into a sequence to show the patterns of historical change. At the same time, a model is generated, with which we can obtain the query similarities by inputting queries and timestamps. Given the above example, we can obtain the query similarity on each day of the week. Moreover, we can monitor how the query similarity changes over time within each week in a daily basis. Also, given two queries and the day of a week, the query similarity can be returned. Then, the process iterates for presenting the results and collecting the click-through data with users interactions. Hereafter, we focus on how to construct the time-dependent query similarity model based on sequences of click-through subgroups.

In order to learn the implicit semantics embedded in the click-through data, we first apply clustering techniques on the data to find the cluster information in each click-through subgroup. When the cluster results are obtained, we then formulate our semantic similarity model by the marginalized kernel technique that can unify both the explicit content similarity and the implicit cluster semantics very effectively. Before the discussion of our semantic similarity model, we first discuss how to cluster the click-through data efficiently. Let us first give a preliminary definition.

Definition 16 Clusters of Click-Through Pages:
Given a click-through subgroup, we can obtain clusters of click-through pages $\Omega=\{c_1,\ c_2,\ \cdots,\ c_k\}$ *by grouping the pages that are similar in semantics, where* k *is determined by some clustering algorithm.* □

In the literature, some clustering methods have been proposed to cluster Web pages in the click-through data using the query-page relation and propagation of similarities between queries and pages [10, 156]. In [10], an agglomerative clustering method is proposed. The basic idea is to merge the most *similar* Web pages and queries iteratively. Originally, the *similarity* is defined based on the overlaps of neighbors in the bipartite graph representation of the click-through data as shown in Figure 6.3.

For the efficiency reason, we adopt the agglomerative clustering method in [10]. In our clustering approach, neighbors in the bipartite graph are assigned with different weights instead of being taken as equal. The intuition is that the strength of the correlation between two query-page pairs may be quite different. For example, the strength of a query-page pair that co-occurs once should not be as equal as a query-page pair that co-occurs thousands of times. Hence, we represent the weights of the neighbors based on the number of times the corresponding query-page pairs co-occur. That is, the weight of a page for a given query is the number of times that page was accessed against the total number of times the corresponding query was issued. Similarly, the weight of a query for a given page is the number of times the query was issued against the total number of times the corresponding page was visited. Then each query is represented as a vector of weighted pages, and each page is represented as a vector of weighted queries. As a result, similarities between pages or queries are calculated based on the cosine similarity measure.

More details about the clustering algorithm can be found in [10]. Note that the clustering algorithm is applied on each of the click-through subgroups. Based on the clustering results, we now introduce the marginalized kernel technique, which can effectively explore the hidden information for similarity measure in a probabilistic framework [73, 138].

Definition 17 Marginalized Kernel: *Assume that a visible variable x is described as $x \in X$, where the domain X is a finite set. Suppose a hidden variable h is described as $h \in H$, where H is a finite set. A joint kernel $K_Z(z, z')$ is defined between the two combined variables $z = (x, h)$ and $z' = (x', h')$. The* ***marginalized kernel*** *in X is defined by taking the expectation with respect to the hidden variables as follows:*

$$K(x, x') = \sum_{h \in H} \sum_{h' \in H} p(h|x) p(h'|x') K_Z(z, z')$$

□

In the above definition, the terms $p(h|x)$ and $p(h'|x')$ are employed to describe the uncertainty of the hidden variables h and h' related to the visible variables x and x', respectively. The marginalized kernel models the probability of similarity between two objects by exploiting the information with the hidden representations. Given the above definition of the marginalized kernel function, we employ it to formulate our time-dependent kernel function for semantic similarity measure of queries as follows.

Definition 18 Time-Dependent Query Semantic Similarity Measure: *Given two queries $\vec{q}$ and $\vec{q}'$, together with a specific timestamp $\vec{t}$, the time-dependent semantic similarity between the two queries is measured by a time-dependent marginalized kernel function $K_T(\vec{q}, \vec{q}'|\vec{t})$ as follows:*

$$\begin{aligned}
&K_T(\vec{q}, \vec{q}'|\vec{t}) \\
&= \sum_{\forall c} \sum_{\forall c'} K_Q\left(Q_{c|t}, Q'_{c'|t}\right) p(c|q, t) p(c'|q', t) \\
&= K_{\cos}\left(q, q'|t\right) \left(\sum_{\forall c} \sum_{\forall c'} \varphi(c, c'|t) p(c|q, t) p(c'|q', t)\right) \\
&= K_{\cos}\left(q, q'|t\right) \left(\sum_{c \in \Omega(t)} p(c|q, t) p(c|q', t)\right) \\
&= \frac{q_t \bullet q'_t}{||q_t|| \times ||q'_t||} \left(\sum_{c \in \Omega(t)} p(c|q, t) p(c|q', t)\right)
\end{aligned}$$

where c and c' are the guessed clusters given the queries, $Q_{c|t} = (q, c|t)$ and $Q'_{c'|t} = (q', c'|t)$. K_Q is a joint kernel, $\varphi(c, c'|t)$ is a function whose value is equivalent to 1 if c and c' are the same and 0 otherwise, and q_t and q'_t are time-dependent query vectors. □

In the above formulation, the joint kernel $K_Q\left(Q_{c|t}, Q'_{c'|t}\right)$ is defined on the two combined query variables as follows:

$$K_Q\left(Q_{c|t}, Q'_{c'|t}\right) = \varphi(c, c'|t) K_{\cos}(q, q'|t),$$

where $\varphi(c, c'|t)$ is a function to indicate whether c and c' are the same cluster of click-through data, and $K_{\cos}(q, q'|t)$ is a time-dependent joint cosine kernel on the two time-dependent query vectors $K_{\cos}(q, q'|t) = \frac{q_t \bullet q'_t}{||q_t|| \times ||q'_t||}$. Note that the query vectors are only computed on the subgroup CTS_i, to which the given timestamp $\vec{t}$ belongs.

From the definition of time-dependent marginalized kernel, we can observe that the semantic similarity between two queries given the timestamp $\vec{t}$ is determined by two factors. One is the time-dependent content similarity measure between queries using the cosine kernel function; another is the likelihood for two queries to be grouped in a same cluster from the click-through data given the timestamp.

6.4 Empirical Evaluation

In this section we conduct a set of empirical studies to extensively evaluate the performance of our time-dependent query semantic similarity model. In the rest of this section, we first describe the dataset used in our evaluation and the experimental setup in our experiments. Then, we show several empirical examples to illustrate the real-world results using our time-dependent framework. After that, we discuss the quality measure metric used in our performance evaluation. Finally, the quality of the time-dependent query similarity model is evaluated under different scenarios.

6.4.1 Dataset

A real click-through dataset collected from Microsoft MSN search engine is used in our experiments. The click-through data contains *15* million records of query-page pairs over *32* days from *June 16, 200*5 to *July 17, 2005*. The size of the raw data is more than *22 GB*. Note that the timestamps for each transaction is converted to the local time using the information about the IP address. In the following experiments, the entire click-through data is partitioned into subgroups based on the user-defined calendar schema and calendar patterns. For instance, given the calendar schema $< hour, day, month >$ with the calendar pattern $< 1, *, * >$, $< 2, *, * >$, $\cdots$, $< 24, *, * >$, the click-through data is partitioned into a sequence of *24* subgroups, where each group consists of the query-page pairs occurred during a specific hour of everyday. Then, the average number of query-page pairs in each group is around *59,400,000*.

6.4.2 Empirical Examples

In this subsection, we present a set of examples of query term similarity evolution over time extracted from the real click-through data collected from MSN search engine. As there are many different types of evolution patterns, here we present some of the representatives.

Figure 6.4 shows the similarities for two query pairs ("kid", "toy") and ("map", "route") on a daily basis in the *32* days. We observe that the similarities changed periodically in a weekly basis. That is, the similarities changed repeatedly: starting low in the first few days of the week and ending high in the weekend. To reflect such time-dependent pattern, we apply our time-dependent query similarity model to the two query pairs. Here the calendar schema and calendar patterns used are $< day, week >$ and $< 1, * >$, $< 2, * >$, $\cdots$, $< 7, * >$. Figure 6.5 shows the time-dependent query similarity measurement for the two

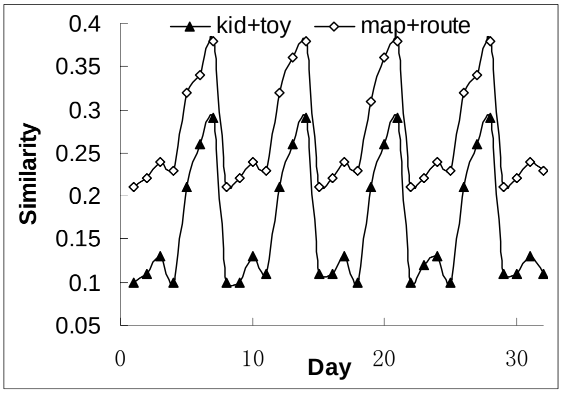


Figure 6.4: Daily-based query similarity evolution

query pairs in Figure 6.4. We can see that the time-dependent query similarity model can efficiently summarize the dynamics of the similarity over time on a weekly basis. However, as shown in Figure 6.6, the *incremented approach* cannot accurately reflect the highs and lows of the similarity values. Note that the calendar schema and calendar patterns used in the model are use-defined with related domain knowledge. With inappropriate calendar schema and calendar patterns, we may not be able to construct accurate time-dependent query similarity models. For instance, for the same query pairs, if we use $< hour$, $day >$ and $< 1, * >$, $< 2, * >$, $\cdots$, $< 24, * >$ as calendar schema and calendar patterns (shown in Figure 6.7). We can see that there are no predictable change patterns, hence there is no *hour of the day* based time-dependent model that can accurately model the similarity.

Figure 6.8 shows the similarity measurement for two query pairs ("weather", "forecast") and ("fox", "news") over one and a half day on hourly basis. We can see that box query pairs have two peak values in every day and this pattern repeatedly occur in the dataset. Based on this observation, we propose to model their similarity using the time-dependent query similarity model with a hourly based calendar patterns. That is, the calendar schema and calendar patterns used

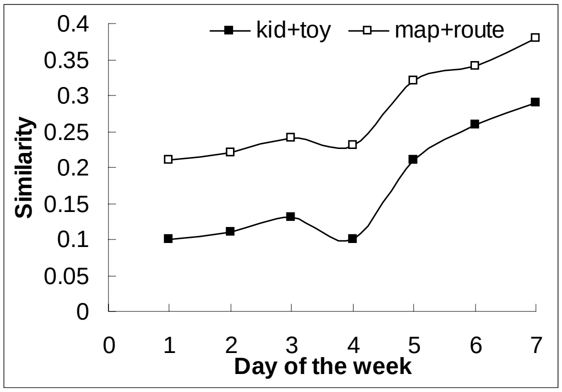


Figure 6.5: Weekly-based time dependent query similarity model

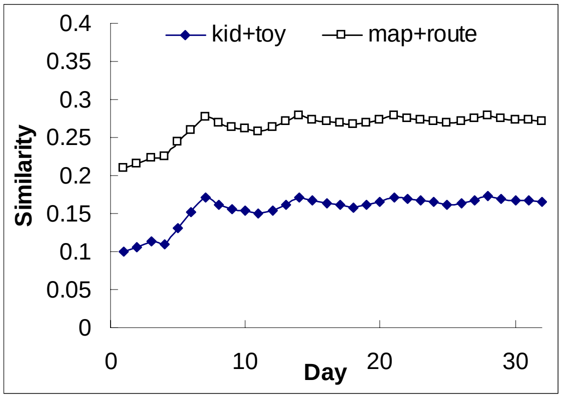


Figure 6.6: Query similarity with incremented approach

are $< hour, day >$ with $< 1, * >$, $< 2, * >$, $\cdots$, $< 24, * >$. Figure 6.9 shows the time-dependent query similarity model. Similarly, Figure 6.10 shows the similarity values calculated using the *incremented approach*, which is clearly not accurate compare to the time-dependent similarity model.

Figure 6.11 shows the similarity measurement of two query pairs (“father”, “gift”) and (“firework”, “show”) on a daily basis. The corresponding similarity values extracted using the *incremented approach* are shown in Figure 6.12. We can see that the time-dependent model cannot be constructed for the two sets of query pairs from the data

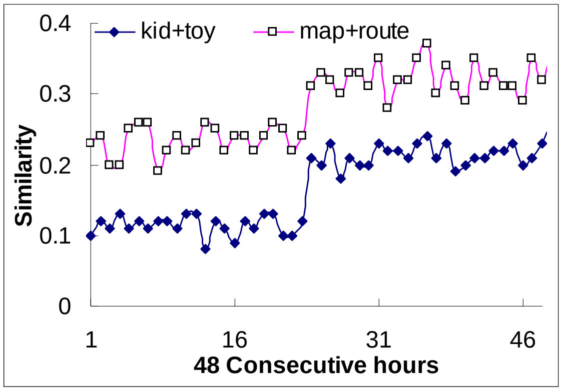


Figure 6.7: Hourly-based query similarity evolution

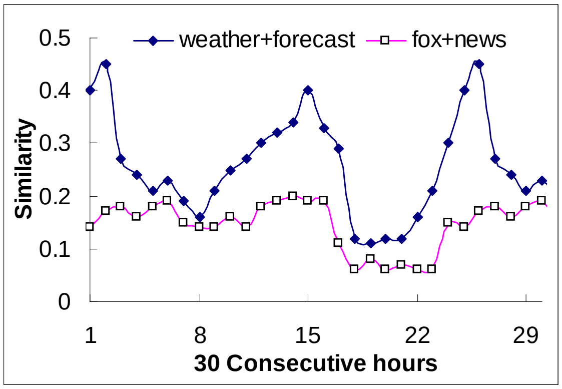


Figure 6.8: Hourly-based query similarity evolution

available in our collection. The reason is that to track event-based query pairs' similarity, e.g., the "father's day" based query pairs' similarity, we need at least years' of click-through data since such events happen only once every year. Note that the previous examples, which can be modeled using the time-dependent model, are within a time interval of *32* days such as weekly based and hour of the day based (our click-through dataset only contains data for *32* days).

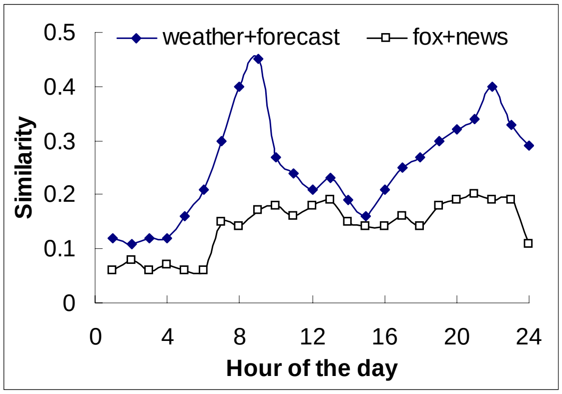


Figure 6.9: Hourly-based time dependent query similarity model

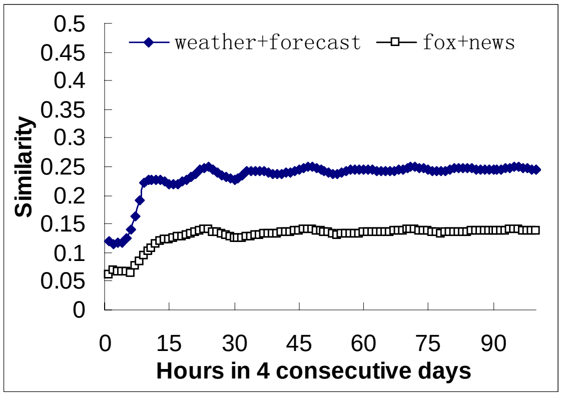


Figure 6.10: Query similarity with incremented approach

6.4.3 Quality Measure

To evaluate the quality of the time-dependent query similarity model, the dataset is partitioned into two parts. The first part consists of a collection of click-through data in the first few days, while the second part consists of the click-through data in the rest of *32* days. Note that the timestamps of click-through data in the first part must be earlier than the timestamps of the click-through data in the second part. The reason is that we will use the first part as training data to construct the time-dependent query similarity model, while the second part is used to evaluate the model. Moreover, partitioning of the click-through

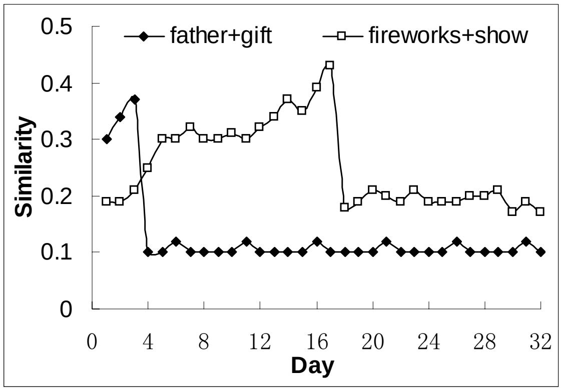


Figure 6.11: Daily-based query similarity evolution

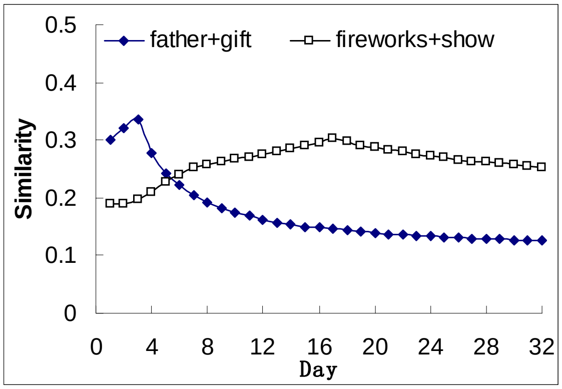


Figure 6.12: Query similarity with incremented approach

dataset also depends on the user-defined *calendar schema* and *calendar patterns*. For example, to build a weekly based model, the training data should at least cover a time duration of one week; a yearly based time-dependent model cannot be constructed using click-through data of a few days.

Once the time-dependent query similarity model is constructed, given a query pair, the similarity can be obtained by matching the corresponding *calendar patterns* in the model. For example, with the weekly based query similarity model as shown in Figure 6.5, the query similarity between "kid" and "toy" can be derived based on the day of the week. We call the similarity derived from the model as *predicted similarity*.

Then, the predicted similarity value is compared with the exact similarity value calculated using the actual dataset. For example, with a weekly based similarity model constructed using the dataset in the first two weeks, the query similarity on the third Monday can be predicted, denoted as S'. Then, the exact similarity is calculated with the dataset in the third Monday. Given the predicted similarity value S' and the exact similarity value S, the *accuracy* of the model is defined as $\frac{|S-S'|}{S}$, where $|S - S'|$ is the absolute difference between the two values. Similarly, for the incremented approach, the same definition of accuracy is used so that we can compare the two approaches.

In the following experiments, a set of *1000* representative query pairs is selected from the query page pairs that have similarities larger than *0.3* in the entire click-through data. Some of them are the top queries in the week or month, some are randomly selected, while others are selected manually based on the related real world events such as "father's day" and "hurricane". Note that the accuracy values shown below are the average accuracy values of all the testing query pairs.

6.4.4 Performance Evaluation

To evaluate the accuracy of the time-dependant query similarity model, three sets of experiments have been done. Firstly, the sizes of the data collections that are used to construct and test the time-dependant query term similarity model are varied. For example, we use the first twenty days as training data and use the eleven days left as testing data or we use the first thirty days as training data and use the last day left as testing data, etc. Note that as the size of the testing data increases, the distance between the training data and test data increases as well. Secondly, only the size of the data collection that is used to constructing the time-dependent model is varied. Whereas the testing data is always the other day followed the dataset being used. For example, we use the first twenty days as training data and use data in the 21^{st} day as testing data. Thirdly, the *distance* between the training data and testing data is varied while the sizes of the training data and testing data are fixed. Note that the *distance* between the two data collections is the distance between the latest query-page pairs in the two collections. For instance, we can use the first twenty days as training data and use data in the 21^{st} day as testing data for the case where *distance* is *1*. If the *distance* is *2*, then data in the 22^{nd} is used for testing. Note that all possible combinations of training and testing data that satisfy the distance constraint are used and the average accuracy values are presented. In the following experiments, if not specified, the calendar schema $<$ *hour*, *day*, *month* $>$ is used with the calendar pattern $< 1, *, * >$, $< 1, *, * >$, $\cdots$, $< 24, *, * >$.

Table 6.2 shows the quality of the time-dependent query similarity model by varying the sizes of data that are used for constructing the model and testing the model. We can see that when the size of the training data increases and size of the testing data decreases, the accuracy of the time-dependent model increases as well. When the sizes of the training and testing data are similar, the accuracy can be as high as 87.3%. Note that here all the click-through data in the *32* days are used. We use the first part of the data as training data and the rest as testing data. The reason behind may be that when the training is not large enough to cover all the possible patterns, then the time-dependent model may not be able to produce accurate results.

Figure 6.13 shows the quality of the time-dependent query similarity model by varying the size of data that is used for construction the model and fixing the size of data that is used for testing to *1*. Three different calendar schemas and calendar patterns are used as well. We

$\|Trainingdata\|$	$\|Testingdata\|$	$Accuracy$
10	22	0.784
15	17	0.873
20	12	0.892
25	7	0.921
30	2	0.968

Table 6.2: Quality of the time-dependent model (1)

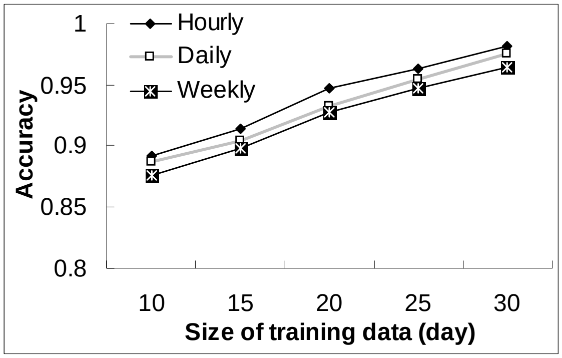


Figure 6.13: Quality of the time-dependent model (2)

can see that when the size of the training data increases, the accuracy of the time-dependent model increases as well. This fact is just as we expected: as the size of the training data increases, performance of the model is expected to increase.

Figure 6.14 shows how the quality of the time-dependent query similarity model changes by varying the time distance between the data collection that is used for testing and the data collection used for constructing the model. For example, when the distance is *1* and the training data size is *10*, we summarize all the accuracy values that use the i to *10+i* days as training and use the *10+1+i* as testing. We can see that when the distance increases, the accuracy of the time-dependent model decreases. At the same time, when the size of the training data increases, with the same distance, the accuracy value

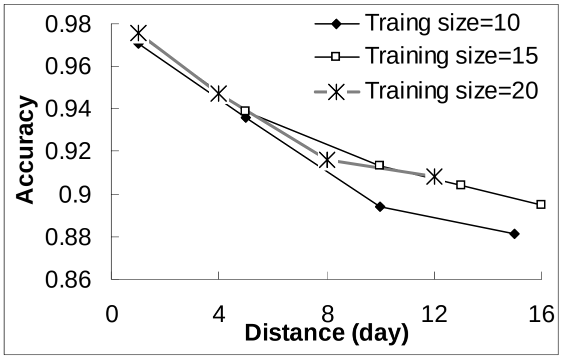


Figure 6.14: Quality of the time-dependent model (3)

may increase. The reason behind this set of data shows that the time-dependent model is more accurate if the most recent data is incorporated as the time-dependent model may be modified.

Moreover, we implemented an incremented query similarity model and compare the prediction accuracy with the time-dependant approach. Note that for the two approaches both the data that are used for building the model and the data that are used for testing are the same (The first part of the data is used for training and the rest is used for testing). In the following experiments, three calendar schema and calendar pattern pairs are used. The calendar schema and calendar patterns are $< hour, day, month >$ with $< 1, *, * >$, $< 2, *, * >$, $\cdots$, $< 24, *, * >$; $< hour, day, month >$ with $< *, 1, * >$, $< *, 2, * >$, $\cdots$, $< *, 31, * >$; and $< day, week >$ with $< 1, * >$, $< 2, * >$, $\cdots$, $< 7, * >$. We use the *1000* sampled query pairs for performance evaluation.

Figure 6.15 shows the comparison of quality about the similarity values obtained using the incremented approach and the time-dependent model. Note that the size of the training data is varied from 1/4 of the dataset to 7/8 of the dataset as well, while the rest is used for testing. We can see that when the intervals in the calendar schema become larger, the quality of the time-dependant model decreases. This is because we only use a click-through data of *32* days, which can produce

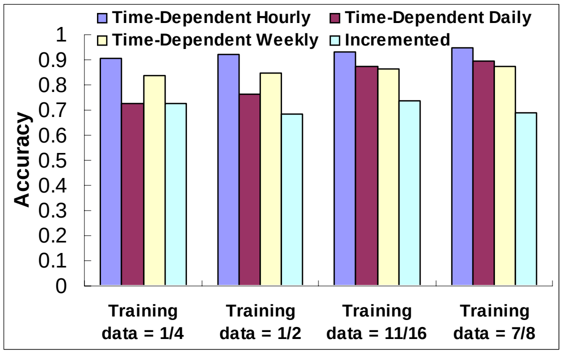


Figure 6.15: Quality of the time-dependent model (4)

satisfactory results with the hourly and daily based calendar patterns. Yet, the quality of the time-dependent model is generally better than the incremented approach. Moreover, we observe that for some calendar schema based time-dependent query similarity model, the accuracy of the model decreases dramatically when the size of the training data decreases, especially for the daily based calendar schema. The reason is that the size of our data collection is not large enough, thus when the size of the training data decreases it cannot cover every possible day in one month (requires at least 31 days of training data).

6.5 Discussions and Future Work

The experiments show that for most query pairs, the similarities are time-dependent and the time-dependent model can produce more accurate similarity values compared to the incremented approach. Besides the time dimension that affects the similarities between queries, there are other factors such as user groups, locations, and topic context, etc. In this chapter, we have focused on incorporating the time dimension. In the future work, we will incorporate other factors mentioned above into the query similarity model. Two extended models are presented as follows.

Personalized time-dependent query similarity model: Beside the time dimension, user groups play an important role in determining the similarities between queries. This is based on the observation that different users have different search habits and have different query vocabularies [51]. For example, some users search about news and business information in the morning and entertainment information in the night, while others may have the reverse habits. Also, to describe the same object or event, people come from different background usually use different query terms. The personalized time-dependent query similarity model is to combine the user information together with the temporal information to build an accurate query similarity model that can be used for improving the personalized search experience.

Spatial-temporal-dependent query similarity model: Similar to the user groups, the spatial location [146] of the queries and Web pages may affect the similarities between queries. For example, for the same object, users in the United States may have different information need compared to users in Asia. Also, contents of Web pages that are created by people in the United States may use different vocabularies compared to those created by people from Asia. By combining the spatial and temporal information, a spatial-temporal-dependent query similarity model can be constructed. There are different types of locations such as *provider location*, *content location*, *serving location*, and *user location*. With such information, we believe, the spatial-temporal-dependent query similarity model can be used to improve the search experience.

6.6 Summary

With the availability of massive amount of click-through data in current commercial search engines, it becomes more and more important to exploit the click-through data for improving the performance of the search engines. We attempt to extract the semantic similarity information between queries by exploring the historical click-through data collected from the search engine. We realize that the correlation between query terms evolves from time to time in the click-through data, which is ignored in the existing approaches. Different from the previous work, we proposed a time-dependent semantic similarity model by studying the temporal information associated with the query terms in the click-through data. We formulated the time-dependent semantic similarity model into the format of kernel functions using the marginalized kernel technique, which can discover the explicit and implicit semantic similarities effectively. We conducted the experiments on the click-through data from a real-world commercial search engine in which promising results show that term similarity does evolve from time to time and our semantic similarity model is effective in modelling the similarity information between queries. Finally, we observed an interesting finding that the evolution of query similarity from time to time may reflect the evolution patterns and events happening in different time periods.

□ **End of chapter.**

Chapter 7

Online Collaborative Multimedia Retrieval

7.1 Problem and Motivation

Given the difficulty in learning the users' information needs from their feedback, multiple rounds of relevance feedback are usually required before satisfactory results are achieved. As a result, the relevance feedback phase can be extremely time-consuming. Moreover, the procedure of specifying the relevance of images in relevance feedback is usually viewed as a tedious and boring step by most users. Hence, it is required for a CBIR system with relevance feedback to achieve satisfactory results within as few feedback steps as possible, preferably in only one step. Despite previous efforts to accelerate relevance feedback using active learning techniques [136], traditional relevance feedback techniques are ineffective when the relevant samples are scarce in the initial retrieval results. From a long-term learning perspective, log data of accumulated users' relevance feedback could be used as an important resource to aid the relevance feedback task in CBIR. Although there have been a few studies carried out on the exploitation of users' log data in document retrieval [3, 33], little research effort has been dedicated to the relevance feedback problem in CBIR [57]. To our best knowledge, there has been no comprehensive work on integrating log of users' feedback into the learning process of relevance feedback in CBIR. Several recent studies related to our work are either too heuristic or lacking empirical evaluations from real-world users [50, 49, 167]. For example, the work in [49] suggested learning a semantic space by mining the relevance feedback log in CBIR. However, only the positive feedback was considered; the negative feedback examples, which can also be informative to users' information needs, were ignored.

In this chapter we present a novel log-based relevance feedback framework for integrating the log data of users' relevance feedback with regular relevance feedback for image retrieval. We refer to the multimedia retrieval approach based on users' log data as "Collaborative Multimedia Retrieval" (CMR) and the log-based relevance feedback retrieval scheme as the "Online Collaborative Multimedia Retrieval" scheme. In our online CMR framework, we compute the relevance information between query images and images in the database using both the log data and the low-level features of images, and combine them to produce a more accurate estimation of relevance score. In order to make the learning algorithm more robust to erroneous log data in real-world applications, we propose a novel support vector machine (SVM) algorithm, named Soft Label SVM, to tackle the noisy data problem.

The rest of this chapter is organized as follows. Section 7.2 provides an overview of our framework for the log-based relevance feedback problem, followed by a formal definition and a unified solution for the problem. Section 7.3 gives a background review of SVMs from the regularization perspective and presents the Soft Label SVM that will be used to solve the log-based relevance feedback problem. Section 7.4 presents a log-based relevance feedback algorithm based on the Soft Label SVM technique. Section 7.5 discusses our experimental testbed and the methodology for performance evaluation of the log-based relevance feedback algorithm. Section 7.6 describes our empirical results for the log-based relevance feedback algorithm. Section 8.6 addresses the limitation of our scheme and the challenging problems for our algorithm, as well as the possible solutions in our future work. Section 8.7 concludes this work.

7.2 A Log-based Relevance Feedback Framework

7.2.1 Overview of Our Framework

We first give an overview of our proposed framework for log-based relevance feedback that systematically integrates the log data of users' relevance judgments with regular relevance feedback for image retrieval. Fig. 7.1 shows the architecture of the proposed system. First, a user launches a query in a CBIR system for searching desired images in databases. Then, the CBIR system computes the similarity between the user query and the image samples in database using the low-level image features. Images with high similarity measure are returned to the user. Next, the user judges the relevance of the initially returned results and submits his or her judgements to the CBIR system. A relevance feedback algorithm refines the initial retrieval results based on the user's relevance judgments, and returns an improved set of results to the user. Typically, a number of rounds of users' relevance feedback are needed to achieve satisfactory results.

Unlike traditional relevance feedback, we propose a unified framework that combines the feedback log with the regular relevance feedback. In Fig. 7.1, we see that the online relevance feedback from users is collected and stored in a log database. When feedback log data is unavailable, the log-based relevance feedback algorithm behaves exactly like a regular relevance feedback algorithm, which learns the correla-

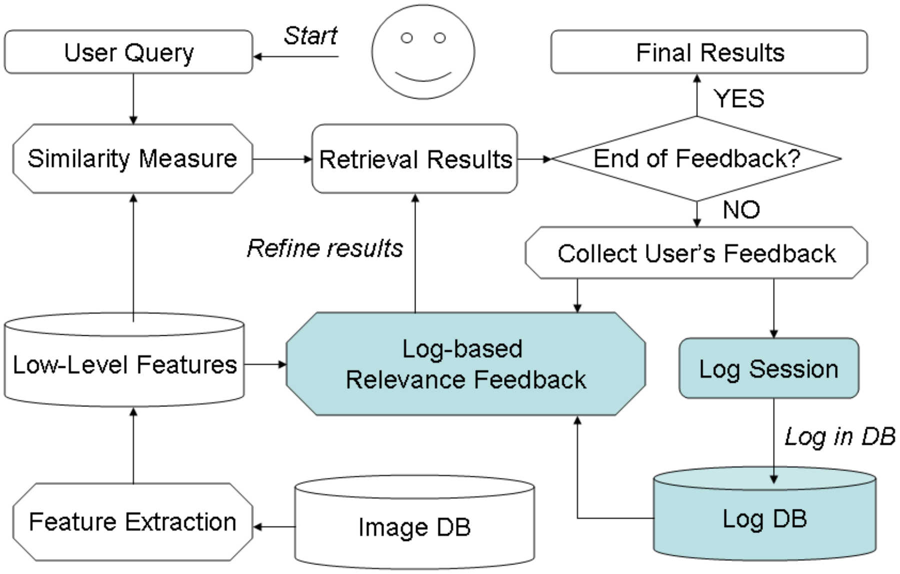
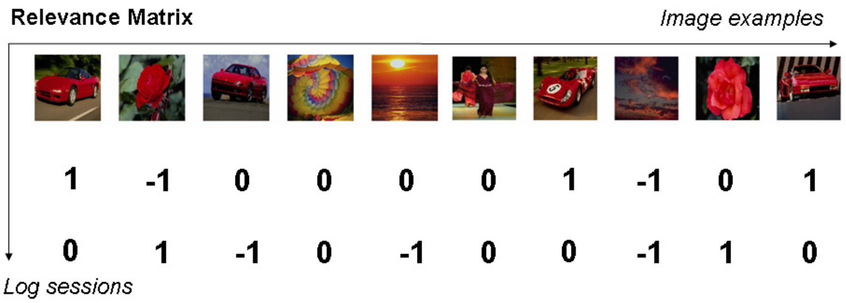


Figure 7.1: The architecture of our proposed system

tion between low-level features and users' information needs through the feedback image examples. When feedback log data is available, the algorithm will learn such a correlation using both the feedback log data and the online feedback from users. Thus, the log-based relevance feedback scheme is able to accomplish the retrieval goal in only a few iterations with the assistance from the log data of users' feedback.

7.2.2 Formulation and Definition

Before formally describing the problem of log-based relevance feedback, we need to systematically organize the log data of users' feedback. Assume a user labels N images in each round of regular relevance feedback, which is called a *log session* in this chapter. Thus, each log session contains N evaluated images that are marked as either "relevant" or "irrelevant". For the convenience of representation, we construct a relevance matrix ($\mathbf{R}$) that includes the relevance judgements from all log sessions. Fig. 7.2 shows an example of such a matrix. In this figure, we see that each column of a relevance matrix represents an image example in the image database, and each row represents a log session from the log database. When an image is judged as “relevant” in a log session, the corresponding cell in matrix $\mathbf{R}$ is assigned to the value $+1$. Similarly, -1 is assigned when an image is judged as “irrelevant”. For images that are not judged in a log session, the corresponding cells in $\mathbf{R}$ are assigned to zero values.



1	-1	0	0	0	0	1	-1	0	1
0	1	-1	0	-1	0	0	-1	1	0

Figure 7.2: The relevance matrix for representing the log information of user feedback. Each column of the matrix represents an image example in the image database and each row of the matrix corresponds to a log session in the log database.

Based on the above formulation, we now define the log-based relevance feedback problem. Let us first introduce the following notation:

- $\mathbf{q}$: a user query
- N_l: the number of labeled images for every log session
- N_{img}: the number of image samples in the image database
- N_{log}: the number of log sessions in the log database

To retrieve the desired images, a user must first present a query $\mathbf{q}$, either by providing a query image or by drawing a sketch picture. Let $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{N_{img}}\}$ denote the identity of images in the image database. Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_{img}})$ denote the image database, where each $\mathbf{x}_i$ is a vector that contains the low-level features of the image $\mathbf{z}_i$. Let $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_{N_{log}})^T$ denote the log data in the log database, where each $\mathbf{r}_i$ contains relevance judgements in the i-th log session. Let $\mathcal{L} = \{(\mathbf{z}_1, y_1), (\mathbf{z}_2, y_2), \ldots, (\mathbf{z}_{N_l}, y_{N_l})\}$ be the collection of labeled images acquired through the online feedback for a user. Then, the definition of a log-based relevance feedback problem can be given as follows:

Definition 19 (Log-based Relevance Feedback) *A log-based relevance feedback problem for image retrieval is to look for a relevance function $f_{\mathbf{q}}$ that maps each image sample $\mathbf{z}_i$ to a real value of relevance degree within 0 and 1,*

$$f_{\mathbf{q}} : \mathcal{Z} \longmapsto [0, 1],$$

based on the feature representation of images $\mathbf{X}$, the log data of users' feedback $\mathbf{R}$, and the labeled images $\mathcal{L}$ acquired from online feedback.

According to the above definition, both the low-level features of the image content, i.e., $\mathbf{X}$, and the log data of users' feedback, i.e, $\mathbf{R}$, should be included to determine the relevance function $f_{\mathbf{q}}$. Meanwhile, to reduce the number of iterations of online relevance feedback, a good learning algorithm should require only a small number of labeled image examples from the online relevance feedback, i.e., $|\mathcal{L}|$.

7.2.3 Solution to the Problem

Given that the relevance function depends on both $\mathbf{R}$ and $\mathbf{X}$, a simple strategy is to first learn a relevance function for each of these two types of information, and then combine them through a unified scheme. Let $f_{\mathbf{R}}(\mathbf{z}_i)$ denote a relevance function based on the log data of users' feedback, and $f_{\mathbf{X}}(\mathbf{z}_i)$ denote a relevance function based on the low-level features of the image content. Both of them are normalized to $[0, 1]$ respectively. Then, the overall relevance function can be the combination of these two functions as follows:

$$f_{\mathbf{q}}(\mathbf{z}_i) = \frac{1}{2}(f_{\mathbf{R}}(\mathbf{z}_i) + f_{\mathbf{X}}(\mathbf{z}_i)), \tag{7.1}$$

In the following, we will describe how to acquire the relevance functions $f_{\mathbf{R}}(\mathbf{z}_i)$ and $f_{\mathbf{X}}(\mathbf{z}_i)$ separately.

Let us first consider the log data of users' feedback. When two images have similar content, we would expect different users to express similar relevance judgements for these two images. On the other hand, for two images with dramatically different content, there should be no correlation in their relevance judgments in log data. Hence, to estimate the similarity between two images $\mathbf{z}_i$ and $\mathbf{z}_j$, we suggest a modified correlation function to measure their relevance judgments in the log data, i.e.,

$$c_{i,j} = \sum_k \delta_{k,i,j} \cdot r_{k,i} \cdot r_{k,j} \tag{7.2}$$

where $\delta_{k,i,j}$ is defined as follows:

$$\delta_{k,i,j} = \begin{cases} 1 & \text{if} \quad r_{k,i} + r_{k,j} \geq 0 \ , \\ 0 & \text{if} \quad r_{k,i} + r_{k,j} < 0 \ . \end{cases} \tag{7.3}$$

Note that $\delta_{k,i,j}$ is engaged to remove $(-1, -1)$ pairs among $(r_{k,i}, r_{k,j})$ in the computation of similarity. This is because it is difficult to judge similarity of two images when they both are marked as "irrelevant" to users' information needs. Evidently, image $\mathbf{z}_i$ and image $\mathbf{z}_j$ are relevant when c_{ij} is positive, irrelevant when c_{ij} is negative. When c_{ij} is around zero, it is usually hard to judge if one image is relevant to the other.

Based on the above similarity function, we can develop the relevance function based on the log data. Let $\mathcal{L}^+$ denote the set of positive (or relevant) images in $\mathcal{L}$, and $\mathcal{L}^-$ denote the set of negative (or irrelevant) samples. For an image in the database, we compute its overall similarities to both positive and negative images, and the difference between these two similarities will indicate the relevance of the image to the user's query. More specifically, the overall relevance function can be formulated as follows:

$$f_{\mathbf{R}}(\mathbf{z}_i) = \max_{k \in \mathcal{L}^+} \left\{ \frac{c_{k,i}}{\max_j c_{k,j}} \right\} - \max_{k \in \mathcal{L}^-} \left\{ \frac{c_{k,i}}{\max_j c_{k,j}} \right\} \tag{7.4}$$

Despite its simple form, our empirical studies have shown that the above relevance function is effective in practice [57].

Remark: So far we assume the above relevance function is calculated on the fixed log data. Toward a long-term learning purpose, it is important to develop an incremental method to deal with the new added log session. For the method proposed above, it is natural to provide an incremental solution. For example, we can create a correlation matrix $C_M = [c_{i,j}]_{N_{img} \times N_{img}}$ which marks down the correlation values between images based on the history log data. When a new log session is added to the log database, we can update the element in the correlation matrix as follows:

$$c_{i,j} = c_{i,j} + \delta_{k',i,j} \cdot r_{k',i} \cdot r_{k',j} \tag{7.5}$$

where $\mathbf{r}_{k'}$ is the new log session, $\delta_{k',i,j}$ is defined in (7.3). Note that only the element $c_{i,j}$ satisfying $r_{k',i} \neq 0$ and $r_{k',j} \neq 0$ will be updated.

After obtaining the relevance function on the log data, we can use it in learning the relevance function on the low-level image features. Learning the relevance function on the image features is a standard relevance feedback problem in content-based image retrieval. Dozens of suitable algorithms have been proposed in the literature [61]. Among them, support vector machine (SVM) is one of the most effective techniques in practice. As a state-of-the-art classification technique, SVM enjoys excellent generalization capability which has shown superior performance in many applications. Although it is able to function with small numbers of training samples, the performance of SVM will usually deteriorate significantly when the number of training samples is too small. This is a general issue with any discriminative classifier as pointed out in [99]. Given that the number of labeled samples in $\mathcal{L}$ is small, applying SVM directly to $\mathcal{L}$ may not achieve the desirable performance. One possible solution is to boost the performance using unlabeled samples by the Transductive SVM [69]. However, difficulties such as high training cost [69] and unstable performance prevent its application to the relevance feedback problem.

Hence, we propose enriching training samples by employing the relevance function based on the log data in (7.4). One simple approach is to calculate the relevance scores of image samples to the query target using (7.4), and augment training examples with the image samples that have large relevance scores. Although this approach can be straightforwardly handled by the standard SVM algorithm, it may suffer from performance degradation, providing that image samples with high relevance scores may not be relevant to the targeted query. To deal with this noisy data problem, we propose a novel learning algorithm, named Soft Label Support Vector Machine. Unlike the standard SVMs in which all the training examples are labeled as either "+1" or "-1", our algorithm does not require absolute confidence about the labels of the selected training samples. In fact, the relevance scores of images reflect the uncertainties in determining their labels. Thus, instead of using hard binary labels, we introduce the "soft label" for the training samples that use the relevance scores computed from (7.4). By combining the soft-labeled samples with the labeled samples acquired from the online user feedback, we can train a Soft Label SVM classifier. The final relevance function on the low-level image features will be constructed based on the decision function of the trained classifier. In the following section, we first introduce the background of SVM and then formulate the Soft Label SVM technique in detail.

7.3 Soft Label Support Vector Machines

7.3.1 Overview of Regularization Learning Theory

To provide a rigorous justification of Soft Label Support Vector Machine, we here provide a brief overview of regularization framework and Support Vector Machines.

In a general setting of learning from examples, we are given a training set of l independent and identically distributed observations

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)$$

where $\mathbf{x}_i$ are vectors produced by a generator and y_i are the associated responses by a supervisor. A learning machine estimates a set of approximated functions f to approach the supervisor's responses.

The classical regularization theory has been justified by the significant work of Vapnik's theory [141]. Based on the framework of Vapnik's theory, a general regularization framework is suggested to find the function f via the functionals

$$f = \arg \min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(\mathbf{x}_i, y_i, f) + \lambda \|f\|_K^2, \tag{7.6}$$

where $V(\cdot, \cdot, \cdot)$ is a loss function, and the penalty norm $\lambda \|f\|_K^2$ imposes smoothness conditions on the solution space.

To avoid confusions, we limit our further discussion on classification and retrieval problems. The loss function in (2.4) is also named as the soft margin loss function for SVM classification.

7.3.2 Soft Label Support Vector Machines

According to the regularization framework in (7.6), it is critical to define an appropriate loss function that fits in with the nature of the application. In standard SVMs for classification applications, the given training samples are normally assumed noise-free. When this assumption is not satisfied, the original loss function may not be the best choice. This motivates us to study the Soft Label Support Vector Machines for the cases of noisy labels. To facilitate the following discussion, we denote the regular support vector machine for noise-free cases as "Hard Label Support Vector Machine" (SVM), and the noise-appearing cases as "Soft Label Support Vector Machine" (SLSVM).

Suppose we are given the training data as follows:

$$(\mathbf{x}_1, s_1), (\mathbf{x}_2, s_2), \ldots, (\mathbf{x}_m, s_m)$$

where the label s_i is a real number and $0 < |s_i| < 1$ [1]. In the above setting, the sign of each label s_i, i.e., $sgn(s_i)$, indicates the binary class label of the corresponding sample. The magnitude of label s_i, i.e., $|s_i|$, represents the confidence of the assigned label. We call these labels "soft labels" to distinguish them from the binary labels. Our goal is to learn a reliable SVM classification model from the data points that are "softly" labeled.

A straightforward approach is to convert a Soft Label learning problem into the one with hard labels. However, this will discard the confidence information related to the soft labels, which may significantly degrade the performance of the classifier. In order to develop a more robust scheme for exploiting the information of soft labels, we propose to modify the loss function of SVMs in (2.4). Our first formal definition of the Soft Label loss function is given as follows:

$$V(\mathbf{x}_i, s_i, y_i, f) = |s_i| \cdot (1 - y_i f(\mathbf{x}_i))_+. \tag{7.7}$$

Different from (2.4), the loss term is weighted by $|s_i|$, i.e., the confidence of the assigned label. The larger the confidence $|s_i|$ is, the more important the loss term of the sample will be. We further expand the

[1] If a training sample is given with $s_i = 0$, it will be treated as an unlabeled data instance which is excluded from our learning machine.

loss function defined in (7.7) by including the hard-labeled data, i.e.,

$$\begin{aligned} & V(\mathbf{x}_i, s_i, y_i, f) \\ = & \begin{cases} C_H \cdot (1 - y_i f(\mathbf{x}_i))_+ & \text{if} \quad |s_i| = 1, \\ C_S \cdot |s_i| \cdot (1 - y_i f(\mathbf{x}_i))_+ & \text{if} \quad 0 < |s_i| < 1. \end{cases} \end{aligned} \tag{7.8}$$

In the above definition, we assume the hard-labeled data points correspond to the case when $|s_i| = 1$. Two weight parameters C_H and C_S are introduced to balance the importance between hard-labeled data and soft-labeled data. Usually, we set $C_H > C_S > 0$. This is based on the intuition that the cost of misclassifying a hard-labeled example should be significantly higher than the cost of misclassifying a softly labeled example. By carefully choosing the value of C_S and C_H, our SVM algorithm is able to, on the one hand fully take advantage of the soft-labeled examples to narrow down the best location for the decision boundary, and on the other hand, avoid being misled by the potentially erroneous labels in the soft-labeled data.

Now, assume $f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) - b$. By substituting the definition of loss function in (7.8) into the general framework in (7.6), we have

$$\begin{aligned} \min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + C_H \sum_{i=1}^{l} (1 - y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - b))_+ \\ + C_S \sum_{i=l+1}^{l+m} |s_i|(1 - y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - b))_+ \end{aligned} \tag{7.9}$$

To simplify the above problem, we introduce a slack variable $\xi_i = (1 - y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - b))_+$ for every labeled example (including both hard-labeled instances and soft-labeled instances), which leads to the following optimization problem:

Definition 20 Soft Label Support Vector Machine (SLSVM):

$$\begin{aligned}
\min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_H \sum_{i=1}^{l} \xi_i + C_S \sum_{i=l+1}^{l+m} |s_i|\xi_i && (7.10)\\
\text{subject to} \quad & y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - b) \geq 1 - \xi_i, \\
& \xi_i \geq 0 \ , i = 1, \ldots, l+m \ ,
\end{aligned}$$

where l and m are respectively the number of hard-labeled training data and the number of soft-label ones (with $|s_i| < 1$), C_H and C_S are weight parameters for hard-labeled and soft-labeled training data respectively. For softly labeled examples, $y_i = sgn(s_i)$. □

Note that when all $|s_i| = 0$, the above optimization problem is reduced to a standard SVM.

The solution to the above optimization problems can be found by introducing the Lagrange functional technique, similar to the method of solving standard SVMs [141]. Here we simply state the final result:

$$\begin{aligned}
\max_{\alpha} \quad & \sum_{i=1}^{l+m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l+m} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) && (7.11)\\
\text{subject to} \quad & \sum_{i=1}^{l+m} \alpha_i y_i = 0 \\
& 0 \leq \alpha_i \leq C_H \ , i = 1, 2, \ldots, l \ , \\
& 0 \leq \alpha_i \leq |s_i| C_S \ , i = l+1, l+2, \ldots, l+m \ .
\end{aligned}$$

More details are referred to Appendix A.1. Notice that the upper bounds of the weights α_i for softly labeled examples are proportional to the confidence of their class labels. As a result, the misclassification cost is directly proportional to the confidence of labeling examples. Apparently, this is consistent with our common intuition. Similar to standard SVMs, the optimization problem in (7.11) is a typical QP problem that can be solved effectively by available techniques [101].

7.4 Log-based Relevance Feedback using Soft Label SVM

In Section 7.2 we provide a unified framework to developing a log-based relevance feedback algorithm in general. The key idea is to first identify a relevance function based on the log data of users' feedback, i.e., $f_R(\mathbf{x})$. Then, the log-based relevance function is used to aid the learning task of the relevance function based on the low-level image features, i.e., $f_X(\mathbf{x})$. Finally, these two relevance functions are combined together to rank all the images. Given the erroneous log data, applying traditional techniques to the log-based relevance feedback may be problematic on account of the noise in the data.

To develop an effective log-based relevance feedback algorithm, a modified SVM technique, i.e., the Soft Label SVM, was proposed in the preceding section to attack the noise problem. In contrast to standard SVMs, the Soft Label SVMs incorporates the label confidence into the learning task. In this section, we develop a practical algorithm for log-based relevance feedback using Soft Label SVM, which we refer to as LRF-SLSVM. It can be summarized in four steps as follows:

(1). *Calculate relevance scores $f_R(\mathbf{z})$ for all image samples.* The relevance scores are computed using (7.4) to evaluate the initial relevances of images in the database based on the log data. Despite its simple form, (7.4) is empirically effective.

(2). *Choose training samples with Soft Labels based on their relevance scores.* Image samples with large relevance scores obtained in step (1) will be chosen as pseudo training samples and their relevance scores are normalized to serve as the "soft label" for Soft Label SVM.

(3). *Train a Soft Label SVM classifier on the selected training samples with Soft Labels, i.e., $f_{SLSVM}(\mathbf{z})$.* Given the labeled samples acquired from online feedback and the softly labeled examples acquired in step (2), a Soft Label SVM classifier is trained according to the definition in (A.1).

(4). *Rank images based on the combination of the two relevance functions* $f_R(\mathbf{z})$ *and* $f_{SLSVM}(\mathbf{z})$. The two relevance functions $f_R(\mathbf{z})$ and $f_{SLSVM}(\mathbf{z})$ will first be normalized and then combined together to form the overall relevance function, i.e., $f_q(\mathbf{z}) = f_R(\mathbf{z}) + f_{SLSVM}(\mathbf{z})$.

Fig. 7.3 provides the pseudo-code of the algorithm of log-based relevance feedback by Soft Label SVM, in which the relevance function $f_R(\mathbf{z})$ is represented by $(R_p(\mathbf{z}) - R_n(\mathbf{z}))$. Implementation details of the proposed algorithm will be discussed in the following experimental section.

Table 7.1: Time costs of the proposed schemes (seconds)

Datasets	RF-SVM	LRF-SLSVM		
	$T_{\texttt{SVM}}$	$T_{\texttt{SLSVM}}$	T_{log}	T_{total}
20-Category	5.53	8.09	4.87	12.96
50-Category	13.14	16.85	16.10	32.94

7.5 Experimental Methodology

7.5.1 Overview of Experimental Testbeds

The experimental testbeds and settings are critical to evaluating the performance of log-based relevance feedback algorithms. So far, there is no a benchmark dataset available for log-based relevance feedback problem. Thus, we must design a set of objective and practical experimental testbeds which not only accurately evaluate our algorithms but also adequately facilitate real-word applications.

```
Algorithm: LRF-SLSVM
Input:
  $\mathbf{q}$ /* a query sample by a user */
  $\mathcal{L}$ /* set of labeled training samples */
Variables:
  $\mathcal{S}$ /* set of "Soft Label" training samples */
  $C_H, C_S$ /* regularization parameters in SLSVM */
  $c$ /* correlations or relationships between images */
  $R_p, R_n, f_R$ /* log-based relevance degrees to the query */
  $\Delta$ /* selection threshold for "Soft Label" samples */
  $f_{\mathrm{SLSVM}}$ /* a Soft Label SVM classifier */
  $f_q$ /* the overall relevance function */
Output:
  $\mathcal{R}_{top}$ /* set of most relevant samples */
BEGIN
/* Step. (1) compute log-based relevance functions */
  for each positive $\mathbf{z} \in \mathcal{L}$ {
    for each $\mathbf{z}_i \in \mathcal{Z}$ {
      $c$(i) ← CompRelationship($\mathbf{z}$, $\mathbf{z}_i$); } /* by Equation (7.2) */
    c ← Normalize(c); /* normalize to [0, 1]*/
    $R_p$(i) ← max($R_p$(i), $c$(i)); } /* Init: $R_p(i) \leftarrow -\infty$ */
  for each negative $\mathbf{z} \in \mathcal{L}$ {
    for each $\mathbf{z}_i \in \mathcal{Z}$ {
      $c$(i) ← CompRelationship($\mathbf{z}$, $\mathbf{z}_i$); } /* by Equation (7.2) */
    Normalize(c); /* normalize to [0, 1]*/
    $R_n(i)$ ← max($R_n$(i), $c$(i)); } /* Init: $R_n(i) \leftarrow -\infty$ */
/* Step. (2) select "Soft Label" training samples */
  for each $\mathbf{z}_i \in \mathcal{Z}$ {
    if $R_p(i) - R_n(i) \geq \Delta$, then $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{\mathbf{z}_i\}$; }
/* Step. (3) train a Soft Label SVM classifier */
  $f_{\mathrm{SLSVM}}$ ← Train_Soft_Label_SVM($\mathcal{L}$, $\mathcal{S}$, $C_H$, $C_S$)
  $f_{\mathrm{SLSVM}}$ ← Normalize($f_{\mathrm{SLSVM}}$);
/* Step. (4) rank images based on $f_{\mathrm{SLSVM}}$ and ($R_p - R_n$) */
  $f_R$ ← Normalize($R_p - R_n$);
  $f_q \leftarrow f_{\mathrm{SLSVM}} + f_R$;
  $\mathcal{R}_{top}$ ← Sort_In_Decend_Order($f_q$);
  return $\mathcal{R}_{top}$;
END
```

Figure 7.3: The LRF algorithm by Soft Label SVM

As we known, empirical evaluation of a CBIR system by humans may be somewhat subjective. Hence, it is necessary to develop an automatic mechanism to evaluate the retrieval performance of CBIR. However, several previous studies on log-based relevance feedback simply generate user data through simulations, which may not reflect the true challenges of real-world applications. To address this problem, in our experiment, a testbed is carefully built to allow for the objective evaluation of content-based image retrieval, while maintaining close analogy to real-world applications. In particular, our testbeds include three components: image datasets, low-level image representation, and the collection of users' log data.

7.5.2 Image Datasets

To perform empirical evaluation of our proposed algorithm, we choose the real-world images from the COREL image CDs. There are two sets of data used in our experiments: 20-Category (20-Cat) that contains images from 20 different categories, and 50-Category (50-Cat) that includes images from 50 categories. Each category in the datasets consists of exactly 100 images that are randomly selected from relevant examples in the COREL image CDs. Every category represents a different semantic topic, such as *antique*, *antelope*, *aviation*, *balloon*, *botany*, *butterfly*, *car*, *cat*, *dog*, *firework*, *horse* and *lizard*, etc.

The motivation for selecting images in semantic categories is twofold. First, it allows us to evaluate whether the proposed approach is able to retrieve the images that are not only visually relevant but also semantically similar. Second, it allows us to evaluate the retrieval performance automatically, which will significantly reduce the subjective errors relative to manual evaluations.

7.5.3 Low-Level Image Representation

Image representation is an important step in the evaluation of relevance feedback algorithms in CBIR. Three different sets of features are chosen in our experiments to represent the images: color, edge and texture.

Color features are widely adopted in CBIR for their simplicity. The color feature extracted in our experiments is the color moment. It is close to natural human perception, whose effectiveness in CBIR has been shown in many previous research studies. Three different color moments are used: color mean, color variance and color skewness in each color channel (H, S, and V), respectively. Thus, a 9-dimensional color moment is adopted as the color feature.

Edge features can be very effective in CBIR when the contour lines of images are evident. The edge feature used in our experiments is the edge direction histogram [65]. To acquire the edge direction histogram, an image is first translated to a gray image, and a Canny edge detector is applied to obtain its edge image. Based on the edge images, the edge direction histogram can then be computed. Each edge direction histogram is quantized into 18 bins of 20 degrees each. Hence an 18-dimensional edge direction histogram is employed to represent the edge feature.

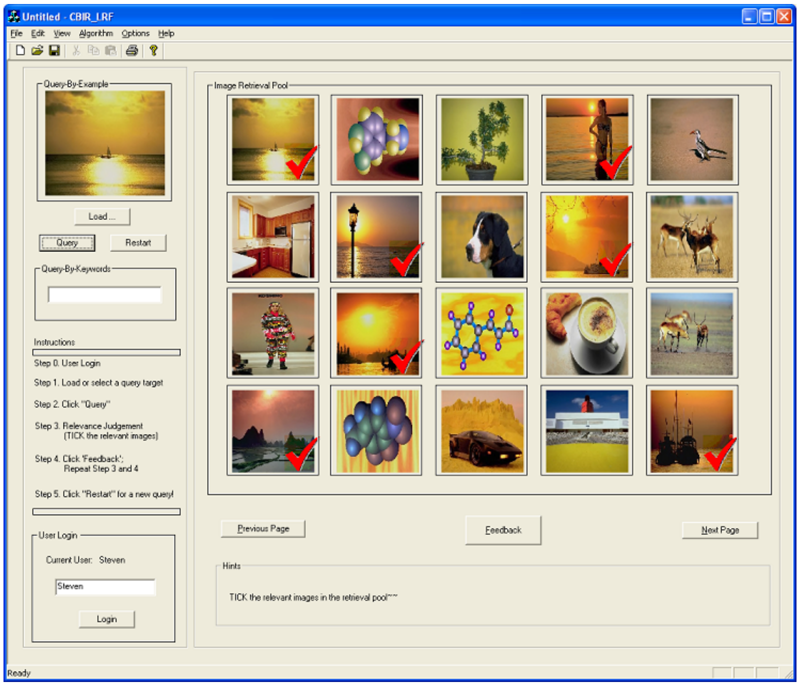
Texture features are proven to be an important cue for image retrieval. In our experiments, we employ the wavelet-based texture technique [91, 126]. A color image is first transformed to a gray image. Then the Discrete Wavelet Transformation (DWT) is performed on the gray image using a Daubechies-4 wavelet filter [126]. Each wavelet decomposition on a gray 2D-image results in four subimages with a $0.5 * 0.5$ scaled-down image of the input image and the wavelets in three orientations: horizontal, vertical and diagonal. The scaled-down image is then fed into the DWT to produce the next four subimages. In total, we perform a 3-level decomposition and obtain 10 subimages in different scales and orientations. One of the 10 subimages is a subsampled average image of the original image, and thus is discarded. For the other 9 subimages, we compute the entropy of each subimage separately. Hence, a wavelet-based texture feature of 9-dimensions in total is computed to describe the texture information of each image.

In sum, a 36-dimensional feature vector is used to represent an image, including 9-dimensional color histogram, 18-dimensional edge direction histogram, and 9-dimensional wavelet-based texture.

7.5.4 Log Data Collection of User Feedback

Collecting the log data of users' feedback is an important step for a log-based relevance feedback scheme. In our experiment we have developed a CBIR system with a relevance feedback mechanism to collect the relevance feedback from real-world users. Fig. 7.4 shows the Graphical User Interface (GUI) of our CBIR system for collecting feedback data. Through the GUI, a user can provide his or her relevance judgements by simply ticking relevant images from the retrieval pool. We describe the details on the collection of the feedback log data and the definition on the format of the log data as follows.

For a retrieval task in CBIR, a user begins a query session by presenting a query example. In our experiment, a user will first randomly select a query image from the image database as the query goal. Then, the user submits the query example to the CBIR system and obtains a set of initial retrieval results from the CBIR system after a query-by-example execution. Based on the retrieval results, the user can tick the relevant images in the retrieval pool. After the relevant samples are ticked in a relevance feedback session, the user can submit his or her judgement results to the CBIR system, in which the feedback results will be stored in the log database. To quantitatively analyze the retrieval performance, we define a log session as the basic unit of the



Instructions

Step 0. User Login

Step 1. Load or select a query target

Step 2. Click "Query"

Step 3. Relevance Judgement
(TICK the relevant images)

Step 4. Click 'Feedback';
Repeat Step 3 and 4

Step 5. Click "Restart" for a new query!

Current User: Steven

Steven

TICK the relevant images in the retrieval pool~~~

Figure 7.4: The GUI of our CBIR system with relevance feedback. A user can simply TICK the relevant images from the retrieval pool to provide his/her feedback. The ticked images are logged as positive samples; others are regarded as negative samples.

log data. Each log session corresponds to a regular relevance feedback session, in which 20 images are judged by the user. Thus, each log session contains 20 labeled images that are marked as either "relevant (positive)" or "irrelevant (negative)."

One important issue with the log data is its noise problem, which is caused by the subjective judgments from the human subjects involved in our study. Given the fact that different users are likely to have different opinions on judging the same image, the noise problem in log-based relevance feedback is inevitable in real-world applications. In order to evaluate the robustness of our algorithm, we collect log data with different amount of noise. The noise of log data is measured by its percentage of incorrect relevance judgments P_{noise}, i.e.,

$$P_{noise} = \frac{\text{Total number of wrong judgements}}{N_l \times N_{log}} \times 100\% \tag{7.12}$$

where N_l and N_{log} stand for the number of labeled examples acquired for each log session and the number of log sessions, respectively.

Table 7.2: The log data collected from users on both datasets

Datasets	Small Noise Log Data		Large Noise Log Data	
	# Log Sessions	P_{noise}	# Log Sessions	P_{noise}
20-Category	100	7.8%	100	16.2%
50-Category	150	7.7%	150	17.1%

In our experiment, 10 users help us collect the log data using our CBIR system. Two sets of log data with different amount of noise are collected on both datasets in the experiment: log data with low noise that contains fewer than 10% of incorrect relevance judgments, and log data with high noise that contains more than 15% of incorrect relevance judgments. Table 7.2 shows the two sets of collected log data for both datasets with different amounts of noise from real-world users. In total, 100 log sessions are collected for the 20-Category dataset and 150 log sessions for the 50-Category dataset. Based on these log data with different configurations, we are able to evaluate the effectiveness, the robustness, and the scalability of our proposed algorithm.

7.6 Experimental Results

7.6.1 Overview of Performance Evaluation

The experiments are designed to answer the following questions:

(1) *Are log-based relevance feedback schemes more effective than traditional relevance feedback methods?* To this end, we compare the per-

formance of log-based relevance feedback algorithms with that of traditional relevance feedback algorithms. Two relevance feedback algorithms are used as our baseline, namely the query expansion approach and the classification approach based on support vector machines.

(2) *Is the proposed algorithm for log-based relevance feedback more effective than other alternatives*? To address this question, we will compare the Soft Label SVM based approach for log-based relevance feedback to other approaches that also utilize the log data to improve the performance of image retrieval. The two methods included in this study are the query expansion based approach and the SVM based approach.

(3) *Is the Soft Label SVM based approach more resilient to noisy log data than the standard SVM based approach?* The noise problem is inevitable in log data. To examine the robustness of the proposed algorithm, we evaluate the performance of the Soft Label SVM based approach against log data with different levels of noise, and compare it with the log-based relevance feedback approach that engages the standard SVM. Since the choice of two weight parameters C_S and C_H can have significant impact on the final retrieval results, we also conduct experiments with different C_S and C_H to see how they affect the robustness of the proposed Soft Label SVM.

7.6.2 The Compared Schemes

In our compared schemes, a simple Euclidean distance measure approach (RF-EU) serves as the baseline method. Two traditional relevance feedback schemes are engaged in our comparisons, i.e., relevance feedback by query expansion (RF-QEX) [103] and relevance feedback by support vector machine (RF-SVM) [136, 162]. In addition to the Soft Label SVM based approach, we also develop two methods for log-based relevance feedback based on our suggested framework by using the traditional query expansion technique (LRF-QEX) and standard SVMs. The details of the compared schemes are given as follows:

Euclidean

This is recorded as a reference of performance comparison. In our approach, Euclidean distances between query images and images in the database are first measured, and images with small distances are then returned to the users. Despite the fact that there have been many other more sophisticated distance measures investigated in CBIR [16], the Euclidean distance scheme is employed in our experiment because of its simplicity and its robustness.

RF-QEX

Query expansion for relevance feedback originates from traditional information retrieval [154, 114]. A lot of different approaches have been proposed to formulate relevance feedback algorithms based on the idea of query expansion [104, 61]. Query expansion can be viewed as a multiple-instance sampling technique [62], in which the returned samples in the next round are selected from the neighborhood of the positive samples of the previous feedback round. Many previous studies have shown that query expansion is effective in relevance feedback for image retrieval [104]. In our experiment, we implement the similar relevance feedback approach in [104] for image retrieval. Specifically, given N_l samples labeled by a user in a relevance feedback round, the images with the smallest Euclidean distances to the N_l positive samples are retrieved to the results. Meanwhile, the negative labeled samples are excluded from the retrieval list if they fall in the selected nearest neighbor of any positive samples.

RF-SVM

Relevance feedback by support vector machine is one of the most popular and promising schemes used in image retrieval [54, 55, 136, 162]. In our experiment, we implement the SVM-based relevance feedback scheme using the Gaussian kernel.

LRF-QEX

Query expansion has shown to be effective in exploiting user query log data in traditional document information retrieval [32]. In our experiment, we extend it to log-based relevance feedback for image retrieval. More specifically, log-based relevance feedback with query expansion can be described as follows:

We first compute the relevance score $f_{\mathbf{R},L}(\mathbf{z}_i)$ for each image $\mathbf{z}_i$ using (7.4). Then, for each image in the database, and for every image $\mathbf{z}_j^+$ that is positively labeled by the user, we measure their Euclidean distance $f_{\mathrm{EU}}(\mathbf{z}_i, \mathbf{z}_j^+)$ based on the low-level image features. The final relevance score $f_{\mathbf{q}}(\mathbf{z}_i)$ for each image $\mathbf{z}_i$ is determined by the combination of $f_{\mathrm{EU}}(\mathbf{z}_i, \mathbf{z}_j^+)$ and $f_{\mathbf{R},L}(\mathbf{z}_i)$, i.e., $f_{\mathbf{q}}(\mathbf{z}_i) = f_{\mathbf{R},L}(\mathbf{z}_i) - \min_j f_{\mathrm{EU}}(\mathbf{z}_i, \mathbf{z}_j^+)$. Images with the largest relevance scores will be returned to the users. As with the query expansion approach for standard relevance feedback, images that are already labeled as negative will be excluded from the retrieval list.

LRF-SLSVM

The algorithm of the log-based relevance feedback by Soft Label SVM is given in Fig. 7.3. To train the Soft Label SVM classifier, similar to standard SVMs, we apply the sequential minimum optimization (SMO) approach [23].

LRF-SVM

To examine the effectiveness and robustness of the Soft Label SVM, we also implement a method for log-based relevance feedback using standard SVMs, which is similar to the algorithm in Fig. 7.3.

7.6.3 Experimental Implementation

The implementation of SVMs in our experiments is based on the public *LIBSVM* library available at [23]. To implement the Soft Label SVM algorithm, we modify the library based on the optimization in (7.11). It is a well-known fact that kernels and their parameters play an important role in the performance of SVMs. In our experiment, the Radial Basis Function (RBF) kernel is used in both the Soft Label SVM and standard SVMs, which is given as $K(\mathbf{x}, \mathbf{x}') = exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$, where γ is a positive constant. The reason for choosing the RBF kernel is that it has been shown to be very effective in multimedia retrieval problems in many previous studies [136, 54]. Besides the kernel selection, the choice of regularization parameters in the standard SVM and the Soft Label SVM is also critical to the retrieval performance. In our experimental implementation, the parameter C in the standard SVM and the two parameters C_H and C_S are chosen empirically using a separate validation dataset.

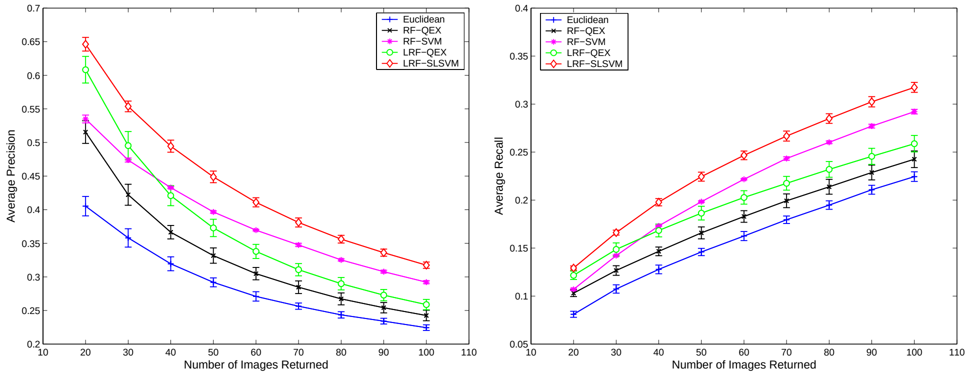
For a retrieval task, it is important to define a suitable metric for performance evaluation. Two metrics are employed in our experiments as follows:

1. **Average Precision**, which is defined as the percentage of relevant images among all the images that have been retrieved; and

2. **Average Recall**, which is defined as the percentage of relevant images of retrieved images among all relevant images in the dataset.

In our experiment, all compared schemes are evaluated on 200 queries randomly selected from the dataset. The reported results of *Average Precision* and *Average Recall* are obtained by taking an average over the 200 queries. For each query, the number of labeled samples acquired from the online user feedback is 10, and the top 100 samples are returned to be evaluated for all compared schemes. To observe the overall performance, *Mean Average Precision* (MAP) is measured on top ranked images, ranging from the top 20 images to the top 100 images. Finally, all the compared schemes are evaluated on both the 20-Category and the 50-category datasets.

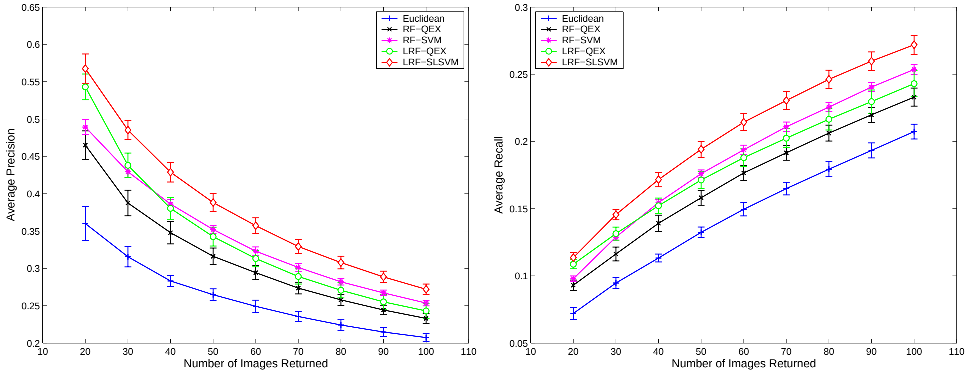
The experimental platform is on Windows and all algorithms are implemented in MS Visual C++ for the purpose of efficiency. The hardware environment of all experiments is a PC machine with a 2.0G Pentium-4 CPU and 512MB memory.



(a) Average Precision

(b) Average Recall

Figure 7.5: Performance evaluation on the 20-Category dataset with small noise log data



(a) Average Precision

(b) Average Recall

Figure 7.6: Performance evaluation on the 50-Category dataset with small noise log data

7.6.4 Effectiveness of Log-based Relevance Feedback

In order to verify the effectiveness of our log-based relevance feedback scheme, we evaluate two log-based relevance feedback algorithms and two traditional relevance feedback algorithms. The algorithms for traditional relevance feedback are the query expansion approach (RF-QEX) and the SVM approach (RF-SVM). The two algorithms for log-based relevance feedback include log-based relevance feedback by query expansion (LRF-QEX) and log-based relevance feedback by Soft Label SVM (LRF-SLSVM). These algorithms are evaluated on the log data with low noise, i.e., 7.8% noise for the 20-Category dataset and 7.7% noise for the 50-Category dataset.

Fig. 7.5 and Fig. 7.6 show the experimental results of the compared algorithms using this log data. The horizontal axis is the number of top ranked images used in evaluation, and the vertical axis is the *Average Precision* and *Average Recall* measured on the top ranked images. As these figures show, it is evident that the two log-based relevance feedback algorithms (LRF-QEX and LRF-SLSVM) substantially outperform the two algorithms with traditional relevance feedback (RF-QEX and RF-SVM). For example, on the 20-Category dataset, the average precision of LRF-QEX algorithm achieves an 18.0% improvement over the regular RF-QEX algorithm on the top 20 images. By contrast, the absolute improvement of the LRF-SLSVM algorithm over the regular RF-SVM algorithm is 20.8% on the top 20 images. With reference to the MAP on average, the LRF-QEX algorithm has an 11.7% improvement over the RF-QEX algorithm, and the LRF-SVM algorithm has a 12.6% improvement over the RF-SVM algorithm.

The results on the 50-Category dataset are similar, but the improvement is slightly smaller than the 20-Category one. This is because the content of the 50-Category is more diverse than the 20-Category one, since the former contains more semantic categories than the latter. As a result, the relevance function based on the log data of users' relevance feedback will less accurately reflect similarity between two images, leading to the degradation in retrieval performance. Nevertheless, we still observe significant improvements with the 50-Category dataset. The average improvement in MAP measure is 8.2% for the LRF-QEX algorithm over the RF-QEX algorithm, and 10.5% for the LRF-SVM algorithm over the RF-SVM algorithm.

Based on the above observations, we conclude that the algorithms for log-based relevance feedback can be expected to outperform the regular relevance feedback schemes.

7.6.5 Performance Evaluation on Small Log Data

In a real-world CBIR application, it may be difficult to collect a large amount of log data, particularly early in the life of a CBIR system. Hence, it is important to evaluate the performance of a log-based relevance feedback algorithm with a small amount of log data. To this end, we evaluate the compared schemes by varying the amount of log

Table 7.3: Performance comparisons (Average Precision) on different amounts of log data on the 20-Category dataset. The baseline algorithm is the regular query expansion algorithm (RF-QEX).

Algorithm	TOP-20	TOP-40	TOP-60	TOP-80	TOP-100	MAP
RF-QEX (Baseline)	0.516±0.017	0.367±0.010	0.305±0.009	0.267±0.009	0.243±0.008	0.332±0.011
RF-SVM	0.535±0.006	0.433±0.002	0.370±0.001	0.325±0.002	0.292±0.002	0.387±0.002
	(+3.8%)	(+18.1%)	(+21.2%)	(+21.7%)	(+20.4%)	(+17.8%)
LRF-QEX (#LS=50)	0.569±0.024	0.395±0.015	0.321±0.013	0.279±0.011	0.251±0.010	0.355±0.015
	(+10.3%)	(+7.8%)	(+5.2%)	(+4.3%)	(+3.6%)	(+6.2%)
LRF-QEX (#LS=100)	0.608±0.020	0.421±0.015	0.338±0.010	0.290±0.009	0.259±0.008	0.374±0.013
	(+18.0%)	(+14.8%)	(+10.9%)	(+8.5%)	(+6.7%)	(+11.7%)
LRF-SLSVM (#LS=50)	0.608±0.009	0.469±0.011	0.391±0.007	0.340±0.005	0.304±0.004	0.416±0.007
	(+17.8%)	(+27.9%)	(+28.2%)	(+27.1%)	(+25.3%)	(+25.8%)
LRF-SLSVM (#LS=100)	0.646±0.010	0.495±0.009	0.411±0.007	0.356±0.006	0.317±0.005	0.438±0.007
	(+25.4%)	(+34.9%)	(+34.8%)	(+33.2%)	(+30.9%)	(+32.4%)

Table 7.4: Performance comparisons (Average Precision) on different amounts of log data on the 50-Category dataset. The baseline algorithm is the regular query expansion algorithm (RF-QEX).

Algorithm	TOP-20	TOP-40	TOP-60	TOP-80	TOP-100	MAP
RF-QEX (Baseline)	0.465±0.019	0.348±0.015	0.294±0.009	0.258±0.007	0.233±0.007	0.313±0.011
RF-SVM	0.489±0.010 (+5.2%)	0.386±0.006 (+11.0%)	0.323±0.006 (+9.7%)	0.282±0.004 (+9.4%)	0.254±0.004 (+8.8%)	0.343±0.006 (+9.5%)
LRF-QEX (#LS=75)	0.509±0.016 (+9.4%)	0.366±0.013 (+5.2%)	0.304±0.010 (+3.2%)	0.264±0.008 (+2.2%)	0.238±0.008 (+2.2%)	0.328±0.011 (+4.3%)
LRF-QEX (#LS=150)	0.543±0.017 (+16.8%)	0.380±0.015 (+9.4%)	0.313±0.011 (+6.4%)	0.271±0.010 (+5.0%)	0.243±0.010 (+4.3%)	0.342±0.012 (+8.2%)
LRF-SLSVM (#LS=75)	0.536±0.016 (+15.2%)	0.407±0.009 (+17.1%)	0.341±0.008 (+16.0%)	0.295±0.007 (+14.6%)	0.262±0.005 (+12.6%)	0.363±0.009 (+15.5%)
LRF-SLSVM (#LS=150)	0.568±0.020 (+22.0%)	0.429±0.013 (+23.3%)	0.357±0.011 (+21.4%)	0.308±0.008 (+19.4%)	0.272±0.007 (+16.7%)	0.381±0.011 (+21.0%)

Table 7.5: Performance comparisons (Average Precision) on the log data with different amounts of noise on the 20-Category dataset. The baseline algorithm is the regular query expansion algorithm (RF-QEX).

Algorithm	TOP-20	TOP-40	TOP-60	TOP-80	TOP-100	MAP
RF-QEX (Baseline)	0.516±0.017	0.367±0.010	0.305±0.009	0.267±0.009	0.243±0.008	0.332±0.011
RF-SVM	0.535±0.006 (+3.8%)	0.433±0.002 (+18.1%)	0.370±0.001 (+21.2%)	0.325±0.002 (+21.7%)	0.292±0.002 (+20.4%)	0.387±0.002 (+17.8%)
LRF-SVM (Low Noise)	0.626±0.010 (+21.3%)	0.474±0.006 (+29.2%)	0.391±0.002 (+28.3%)	0.335±0.003 (+25.5%)	0.298±0.003 (+22.8%)	0.418±0.005 (+25.9%)
LRF-SLSVM$^{\mathrm{SR}}$ (Low Noise)	0.635±0.012 (+23.1%)	0.484±0.007 (+32.1%)	0.401±0.004 (+31.4%)	0.344±0.004 (+28.6%)	0.305±0.003 (+25.8%)	0.427±0.006 (+28.7%)
LRF-SLSVM$^{\mathrm{DR}}$ (Low Noise)	0.646±0.010 (+25.4%)	0.495±0.009 (+34.9%)	0.411±0.007 (+34.8%)	0.356±0.006 (+33.2%)	0.317±0.005 (+30.9%)	0.438±0.007 (+32.4%)
LRF-SVM (High Noise)	0.557±0.021 (+8.1%)	0.433±0.016 (+18.2%)	0.366±0.010 (+20.1%)	0.317±0.009 (+18.6%)	0.283±0.009 (+16.6%)	0.386±0.013 (+17.0%)
LRF-SLSVM$^{\mathrm{SR}}$ (High Noise)	0.584±0.010 (+13.3%)	0.451±0.005 (+23.1%)	0.378±0.002 (+24.1%)	0.327±0.004 (+22.5%)	0.293±0.004 (+20.7%)	0.401±0.005 (+21.3%)
LRF-SLSVM$^{\mathrm{DR}}$ (High Noise)	0.608±0.011 (+18.0%)	0.470±0.009 (+28.3%)	0.398±0.009 (+30.6%)	0.348±0.011 (+30.1%)	0.310±0.009 (+27.8%)	0.421±0.010 (+27.5%)

Table 7.6: Performance comparisons (Average Precision) on the log data with different amounts of noise on the 50-Category dataset. The baseline algorithm is the regular query expansion algorithm (RF-QEX).

Algorithm	TOP-20	TOP-40	TOP-60	TOP-80	TOP-100	MAP
RF-QEX (Baseline)	0.465±0.019	0.348±0.015	0.294±0.009	0.258±0.007	0.233±0.007	0.313±0.011
RF-SVM	0.489±0.010	0.386±0.006	0.323±0.006	0.282±0.004	0.254±0.004	0.343±0.006
	(+5.2%)	(+11.0%)	(+9.7%)	(+9.4%)	(+8.8%)	(+9.5%)
LRF-SVM	0.547±0.015	0.406±0.009	0.337±0.008	0.293±0.007	0.261±0.006	0.363±0.009
(Low Noise)	(+17.6%)	(+16.7%)	(+14.7%)	(+13.8%)	(+11.9%)	(+15.3%)
LRF-SLSVM$^{\mathrm{SR}}$	0.556±0.016	0.423±0.012	0.350±0.010	0.304±0.008	0.271±0.007	0.375±0.011
(Low Noise)	(+19.5%)	(+21.6%)	(+19.1%)	(+17.8%)	(+16.4%)	(+19.5%)
LRF-SLSVM$^{\mathrm{DR}}$	0.568±0.020	0.429±0.013	0.357±0.011	0.308±0.008	0.272±0.007	0.380±0.011
(Low Noise)	(+22.0%)	(+23.3%)	(+21.4%)	(+19.4%)	(+16.7%)	(+21.0%)
LRF-SVM	0.503±0.015	0.385±0.009	0.323±0.006	0.282±0.005	0.251±0.005	0.344±0.008
(High Noise)	(+8.2%)	(+10.6%)	(+9.7%)	(+9.5%)	(+7.9%)	(+9.8%)
LRF-SLSVM$^{\mathrm{SR}}$	0.519±0.018	0.393±0.012	0.328±0.010	0.289±0.008	0.257±0.007	0.352±0.011
(High Noise)	(+11.6%)	(+12.9%)	(+11.4%)	(+12.2%)	(+10.2%)	(+12.2%)
LRF-SLSVM$^{\mathrm{DR}}$	0.530±0.020	0.408±0.013	0.340±0.011	0.295±0.008	0.264±0.007	0.362±0.012
(High Noise)	(+14.0%)	(+17.2%)	(+15.5%)	(+14.5%)	(+13.4%)	(+15.5%)

data. In particular, for each dataset, only half of its log data is used for log-based relevance feedback. This amounts to 50 log sessions for the 20-Category dataset, and 75 log sessions for the 50-Category dataset. The empirical results for the reduced log data are shown in Table 7.3 and 7.4.

According to the two tables, we observe that the log-based relevance feedback algorithm by Soft Label SVM (LRF-SLSVM) achieves a promising improvement even with a limited amount of log data. Most impressively, the mean average precision (MAP) of Soft Label SVM using only half of the log sessions is better than the LRF-QEX approach that uses all the log sessions. For the 20-Category dataset, with only 50 log sessions, the LRF-SLSVM algorithm outperforms the baseline algorithm (RF-QEX) by 25.8% and also enjoys a 6.9% improvement over the regular RF-SVM algorithm. The improvement on the 50-Category dataset is again less than the 20-Category one. But the LRF-SLSVM algorithm still outperforms the RF-QEX algorithm by 15.5% and has a 5.5% improvement over the RF-SVM algorithm with only 75 log sessions.

7.6.6 Performance Evaluation on Noisy Log Data

The presence of noise in the log data is unavoidable when the data is collected from a real-world CBIR application. It is therefore important to evaluate whether a good log-based relevance feedback algorithm is resilient to the noise present in the log data.

In this subsection, we conduct experiments to evaluate the robustness of algorithms on the log data with different levels of noise, meanwhile we compare the performance of SLSVM using different regularization strategies. Two sets of log data on both datasets, with different noise percentages, are employed to evaluate the algorithms. For each of the two datasets, two sets of log data are provided. The noise levels for the 20-Category dataset are 7.8% and 16.2% respectively, and 7.7% and 17.1% respectively for the 50-Category dataset. In addition to varying the amount of noise in the log data, we also conduct experiments for the proposed algorithm LRF-SLSVM with different setup of the two weight parameters C_S and C_H. Two configurations of C_S and C_H are used in this experiment: $C_S = C_H$, which we refer to as (LRF-SLSVM$^{\text{SR}}$), and $C_H > C_S$, which we refer to as (LRF-SLSVM$^{\text{DR}}$).

Tables 7.5 and 7.6 show the comparison results on the datasets with different noise percentages. As expected, performance of the algorithms degrades when a large amount of noise is present in the log data. Compared with other approaches, the Soft Label schemes are more tolerant to the noisy log data. Both the two Soft Label algorithms, i.e., LRF-SLSVM$^{\text{SR}}$ and LRF-SLSVM$^{\text{DR}}$, achieve better performance than the standard SVM algorithm. More impressively, we observe that the performance of the LRF-SLSVM$^{\text{DR}}$ scheme with highly noisy log data is comparable to or better than that of the standard SVM using log data of low noise. Specifically, on the 20-Category dataset, the standard SVM method (LRF-SVM$^{\text{DR}}$) enjoys a 25.9% improvement in MAP over the baseline algorithm under the low noisy log data, while the LRF-SLSVM$^{\text{DR}}$ method achieves a 27.5% improvement even with the highly noisy log data. Similar results can also be observed on the 50-Category dataset. Based on the above observation, we conclude empirically that the Soft Label SVM scheme is more tolerant to the noise than the standard SVM. Finally, comparing the two different configurations of LRF-SLSVM, we observe that LRF-SLSVM$^{\text{DR}}$performs slightly better than LRF-SLSVM$^{\text{SR}}$ for both datasets. This is consistent with our hypothesis, i.e., it is more important to correctly classify the hard-labeled examples than the ones with soft labels.

7.6.7 Computational Complexity and Time Efficiency

Although we have observed significant improvement of our log-based relevance feedback scheme from the above experimental results, it is evident that our scheme requires extra computational cost compared with a regular relevance feedback scheme. Hence, it is necessary to analyze the computational complexity of the log-based relevance feedback scheme and empirically evaluate the time efficiency of our proposed scheme. In our log-based relevance feedback framework, there are two main components that contribute the most to the computational costs. One is the computation of the relevance function on the feedback log data, and the other is the learning of the relevance function on the low-level image features by the Soft Label SVM. It is straightforward to calculate the computational complexity for the former component, which is $\mathcal{O}(N_l \times N_{img} \times N_{log})$. Since N_l, i.e., the number of labeled images acquired from online user feedback, is regarded as a small constant, the time complexity in computing the log information is $\mathcal{O}(N_{img} \times N_{log})$. The major cost for the latter component is in training the SVM; this is determined by the implementation of the optimization problem in the SVM algorithms. In our experiments, the implementations of the SVM algorithms are based on the public libsvm library, for which more detailed analysis of computational cost can be found in [23]. Given that the computational cost for training SVM is highly dependent on the characteristics of the training examples, in the following, we will evaluate the efficiency of the proposed algorithm empirically.

To evaluate the time efficiency, we run 200 executions of relevance feedback with random queries, and record the time costs for both the RF-SVM algorithm and the LRF-SLSVM algorithm. Table 7.1 shows the experimental results of the time costs. The results indicate that extra time costs must be paid for running the LRF-SLSVM compared with the regular RF-SVM scheme. However, the results also suggest that the time costs of the LRF-SLSVM algorithm are still acceptable. For example, for the 50-Category dataset with 150 log sessions, only 32.94 seconds are required for 200 relevance feedback executions, which amounts to only 0.165 seconds for each execution of feedback.

7.7 Limitation and Future Work

Based on the promising results achieved from the extensive evaluations, we can empirically conclude that our log-based relevance feedback scheme is an effective way to improve the traditional relevance feedback techniques by integrating log data of users' relevance feedback. Moreover, the Soft Label SVM algorithm has been demonstrated to be more resilient to the noise problem when solving the log-based relevance feedback problem. However, we must address the limitations of and the challenging issues with our scheme, as well as provide feasible directions for solving these problems in our future work.

The first limitation of our scheme may be the computational complexity problem. Two main computational costs are inherited. One is the relevance computing of log data; and the other is the training cost of Soft Label SVM. For the formal one, the computational cost can be critical when the number of log sessions are huge. Fortunately, our proposed incremental method in (7.5) can partially solve the problem. For the latter one, we can study more efficient decomposition techniques to solve our optimization problem, e.g., the parallel SVMs [45].

Second, it may be possible to learn the relevance function more effectively. In the current scheme, we only consider the classification model in the space of image features. It would be possible to apply the method in the reverse direction by first computing the soft labels from the image features, and then building a classification model in the space of the users' relevance judgement. Furthermore, these two approaches can be integrated together through a co-training algorithm [18].

Third, we realize that the selection of parameter C_H and C_S in the Soft Label SVM algorithm has a major impact on the final retrieval results when deploying the algorithm in the log-based relevance feedback problem. Although our empirical approach for choosing C_H and C_S has resulted in satisfactory performance, we plan to investigate other approaches in principle for tuning these two parameters effectively, e.g., the entire regularization path approach for studying the parameters [46].

Finally, the noise problem could be handled in other ways. For example, to alleviate the negative effect from noisy log data, we can modify the Soft Label SVM by enforcing an upper bound on the error terms in the optimization of the Soft Label SVM.

7.8 Summary

In this chapter we proposed an online collaborative multimedia retrieval framework for bridging the semantic gap with users' context in multimedia information retrieval. Based on the online learning framework, we developed a unified log-based relevance feedback scheme for integrating log data of user feedback with regular relevance feedback for image retrieval. Our solution first computes the relevance function on the log data of user feedback and then combines the relevance information with regular relevance feedback for the retrieval task. In order to address the noisy log data problem in real-world applications, we propose a novel learning algorithm to solve the log-based relevance feedback problem. The proposed algorithm, named Soft Label Support Vector Machine, is based on the solid regularization theory. We have conducted an extensive set of experiments on a sophisticated testbed for evaluating the performance of a number of algorithms on our log-based relevance feedback scheme. The promising experimental results have confirmed that our proposed algorithms are effective in improving the performance of traditional relevance feedback in image retrieval.

The important contributions to the field in this work can be summarized as follows. First, we present a unified framework for studying the log-based relevance feedback problem. To the best of our knowledge, this work is amongst one of only a few pioneering investigations on incorporating both log data of users' feedback and online relevance feedback to improve multimedia retrieval performance. Second, we propose a modified SVM algorithm, i.e., Soft Label SVM, to deal with the problem of noisy log data. Although we employ the Soft Label SVM only in the log-based relevance feedback problem, it can also be applied to other application areas, such as information filtering. Third, we have presented a comprehensive set of experimental procedures for evaluating image retrieval, and for examining various aspects of retrieval algorithms, including effectiveness, efficiency, robustness and scalability.

□ **End of chapter.**

Chapter 8

Offline Collaborative Multimedia Retrieval

8.1 Overview of Our Framework

Recently, there have been several studies on exploring the log data of users' relevance feedback to improve image retrieval [97, 57, 58, 167, 49]. In these studies, the CBIR system collects relevance judgments from a number of users, which is also called "*log data*" in this chapter. In addition to the low-level features, each image is also represented by the users' relevance judgments in log data. Most of these studies hypothesized that when two images are similar in their semantic content, they tend to be either favored or disliked simultaneously by many users. As a result, similar images tend to share similar representation in users' relevance judgments. In [97], several weighting schemes are proposed for the low-level image features that are based on log data. In [50, 48], a manifold learning algorithm is applied to learn a low-dimensional manifold from log data that better reflects the semantic relation among different images. In [57, 58], the log data of users' relevance judgments are used to improve relevance feedback techniques for image retrieval. We refer to the multimedia retrieval approaches based on the metric learned offline from users' log data as "**Offline Collaborative Multimedia Retrieval**".

In this work, we explore the log data of users' relevance judgments in a way that is different from the previous work. Unlike [97] where manually designed weighting schemes based on log data are used to measure similarity of images, in this work, we propose to automatically learn the distance metric for the low-level features from the users' relevance judgements in log data. We hypothesize that, in each user feedback session, when two images are judged as relevant, they tend to be more similar in content than the case when one image is judged as relevant and the other is judged as irrelevant. Thus, our goal is to search for an appropriate distance metric for the low-level features such that the distance in low-level features is consistent with the users' relevance judgments in log data. To this end, we propose the "**Min/Max**" principle, which tries to minimize the distance between similar images and meanwhile maximize the distance between the feature vectors of dissimilar images. Based on this principle, we propose a new algorithm for metric learning, named "**regularized distance metric learning**", in which a regularization mechanism is introduced to improve the robustness of the learning algorithm. The new algorithm can be formulated into an SDP problem [140], and therefore can be solved efficiently by the existing package for SDP, such as SeDuMi [129], and is scalable to the size of log data.

Our work distinguishes from the previous work on exploiting log data for image retrieval in that it deals with the *real-world users* whereas much of the previous research used the synthesized log data in its study. In particular, we try to address the following challenging issues with the *real* log data:

- *Image retrieval with modest-sized log data.* Most previous studies assume that large amount of log data are available, and do not consider the scenarios when the size of log data is limited. Developing retrieval techniques for modest-sized log data is important, particularly when a CBIR system is in its early development and has not accumulated large numbers of relevance judgments from users. It is also important when the target images are not popular and are only equipped with a small number of users' relevance judgments.

- *Image retrieval with noisy log data.* Most previous studies assume that log data are clean and contain no noise. This is an unrealistic assumption given that users' relevance judgments are subjective and real-world users could make mistakes in their judgments. In our experiments with real-world users, we usually observed a number of erroneous relevance judgments, ranging from 5% to 15% of all judgments. As will be shown later in the empirical study, the noise in users' relevance judgments can significantly degrade the retrieval accuracy of a CBIR system.

- *Efficiency and scalability.* Most previous studies emphasize the effectiveness of their algorithms on improving CBIR. Few of them examine the efficiency and scalability of their algorithms. The issue of efficiency and scalability is extremely important for this technique to be practical, particularly when we have to deal with large-sized log data.

The rest of this paper is arranged as follows: the next section discusses the related research. Section 3 describes the proposed regularized metric learning algorithm. Section 4 explains our experimental methodology. Section 5 presents the experimental results. Section 6 discusses the limitation and future work. Section 7 concludes this work.

8.2 Motivation

This work is related to previous studies on utilizing users' log data to enhance content-based image retrieval. It is also related to the research on distance metric learning. We will review the previous work on using log data first, followed by the review of metric learning algorithms.

Users' log data have been utilized in the previous work [57] to improve online relevance feedback for CBIR. In [57], the users' relevance judgments in log data is used to infer the similarities among images. For online retrieval, a set of relevant and irrelevant images are first obtained through the solicitation of users' relevance judgments. Then, based on the log data, images that are most similar to the judged ones are added to the pool of labeled examples, including both relevant and irrelevant images. A discriminative learning model, such as support vector machines (SVM) [21], is trained with the expanded pool of labeled images to improve the retrieval accuracy. This work differs from ours in that it requires online feedback from users, while our algorithm focuses on improving the accuracy of the initial around of image retrieval. Another recent research related to our work is to apply manifold learning to image retrieval [50, 48]. Their work has considered using log data for both CBIR with online feedback and CBIR without online feedback. Using the Laplacian Eigenmap [13], they constructed a low-dimensional semantic space for the low-level image features using log data. Given the complicated distributions of image features, constructing a robust manifold for image features usually requires a large number of training data. In fact, according to our experiments, their algorithm works well when large numbers of users' relevance judgments are available. Its advantage appears to fade away when the size of log data is small. Finally, there are studies on designing weighting schemes for low-level image features based on log data [97]. In [97], weighting schemes, similar to the TF.IDF methods in text retrieval [113], have been proposed and computed based on the log data of users' relevance judgments.

Another group of related work is the learning of distance metric [83, 153]. One of the well-known research on this subject is [153], which learns a distance metric under pairwise constraints. As it serves as the baseline in this study, we briefly describe it here.

Let $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ be the data collection where n is the number of data points in the collection. Each $\mathbf{x}_i \in \mathbb{R}^m$ is a feature vector where m is the number of features. Let $\mathcal{S}$ be the set that contains pairs of similar data points, and $\mathcal{D}$ be the set that contains pairs of dissimilar data points. More precisely, we have

$$\begin{aligned}\mathcal{S} &= \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are likely to belong to the same class}\}\\ \mathcal{D} &= \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are unlikely to be in the same class}\}\end{aligned} \tag{8.1}$$

Let $\mathbf{A} \in \mathbf{S}^{m\times m}$ be the distance metric to be learned, which is a symmetric matrix of size $m \times m$. Then, for any two vectors $\mathbf{x}, \ \mathbf{y} \in \mathbb{R}^m$, their distance is expressed as:

$$d_{\mathbf{A}}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T\mathbf{A}(\mathbf{x}-\mathbf{y})} = \mathrm{tr}(\mathbf{A}\cdot(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T) \tag{8.2}$$

where product "·" is a point wise matrix multiplication, and "tr" stands for the trace operator that computes the sum of diagonal elements of a matrix.

$\mathbf{A}$ is a valid metric as long as the distance between any two data points is non-negative and satisfies the triangle inequality. This requirement is formalized as the positive semi-definite constraint for matrix $\mathbf{A}$, i.e., $\mathbf{A} \succeq 0$ [140]. Furthermore, matrix $\mathbf{A}$ should be symmetric, namely $\mathbf{A} = \mathbf{A}'$. Note when $\mathbf{A}$ is an identity matrix $\mathbf{I}_{m\times m}$, the distance in Eqn. (8.2) becomes

$$d_{\mathbf{A}}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T\mathbf{I}(\mathbf{x}-\mathbf{y})} = \sqrt{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})}$$

Thus, we go back to the Euclidean distance.

Given the pair wise constraints in (8.1), [153] formulated the problem of metric learning into the following convex programming problem:

$$\begin{aligned} \min_{\mathbf{A}} & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 \\ \text{s. t.} & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 \geq 1 \\ & \mathbf{A} \succeq 0 \end{aligned} \tag{8.3}$$

In above, optimal metric $\mathbf{A}$ is found by minimizing the sum of squared distance between pairs of similar data points, and meanwhile satisfying the constraint that the sum of squared distance between dissimilar data points is larger than 1. In other words, this algorithm tries to minimize the distance between similar data points and maximize the distance between dissimilar data points at the same time. This is consistent with our Min/Max principle discussed in the introduction section.

The algorithm in (8.3) has been shown to be successful on several machine learning testbeds [153]. But one potential problem with this method is that it does not address the issue of robustness, which is important when training data are noisy or the amount of training data is limited. Our algorithm is able to improve the robustness of metric learning by introducing a regularizer into the objective function, which is similar to the strategy used in large margin classifiers [21]. Furthermore, the optimization problem in (8.3) may not be solved efficiently since it does not fall into any special class of convex programming, such as quadratic programming [41] and semi-definite programming [140]. In contrast, the proposed algorithm belongs to the family of semi-definite programming, which can be solved much more efficiently.

8.3 Regularized Metric Learning and Its Application

As it is discussed in the introduction section, the basic idea of this work is to learn a desired distance metric in the space of low-level image features that effectively bridges the semantic gap. It is learned from the log data of users' relevance feedback based on the Min/Max principle, i.e., minimize/maximize the distance between the feature vectors of similar/dissimilar images. Log data, in this study, consist of a number of log sessions and each session corresponds to a different user query. In each log session, a user submits a query image to the CBIR system. After the initial results are retrieved by the CBIR system, the user provides relevance judgments for the top ranked images (i.e., 20 images in our experiment). To exploit the metric learning algorithm in (8.3) for log data, we convert binary relevance judgments into pair-wise constraints as in (8.1). In particular, within each log session, images judged as relevant are regarded as similar to each other, and each dissimilar pair will consist of one relevant image and one irrelevant image. Thus, for each user query q, we have a set $\mathcal{S}_q$ for pairs of similar images and a set $\mathcal{D}_q$ for pairs of dissimilar images. Based on this treatment, we can now apply the framework in (8.3) to learn a distance metric $\mathbf{A}$ for low-level image features, i.e.,

$$\begin{aligned} \min_{\mathbf{A}} & \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 \\ \text{s. t.} & \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 \geq 1 \\ & \mathbf{A} \succeq 0 \end{aligned} \tag{8.4}$$

where Q stands for the number of sessions in log data.

Remark. One natural question regarding to the above treatment is that, although two images are judged as relevant by a user, they may still differ in many aspects. There are images that are judged differently by multiple users due to their different information needs. For example, two images could be judged both to be relevant by one user, and but only one being relevant by another user. Hence, it is questionable to treat relevant images as a similar pair. To answer this question, we need to understand that similar pairs $\mathcal{S}_q$ and dissimilar pairs $\mathcal{D}_q$ play different roles in (8.4). The pairs in dissimilar set $\mathcal{D}_q$ are used to form the constraint and the pairs in the similar set $\mathcal{S}_q$ are used to form the objective. Thus, a solution $\mathbf{A}$ to (8.4) must satisfy the constraint first before it minimizes the objective function. As a result, (8.4) only ensures the image pairs in $\mathcal{D}_q$ to be well separated in the feature space, but it does not guarantee that all the image pairs in $\mathcal{S}_q$ are close to each other. In other words, what is implied under the formulism in (8.4) is:

- When two images are judged as relevant in the same log session, they **could** be similar to each other,
- When one image is judged as relevant and another is judged as irrelevant in the same log session, thy **must** be dissimilar to each other.

Clearly, the above assumption is closer to reality than the original one.

One problem with the formulism in (8.4) is that its solution may not be robust when the amount of log data is modest or the relevance judgments in log data are noisy. To enhance the robustness of metric learning, we form a new objective function for distance metric learning that takes into account both the discriminative issue and the robustness issue, formally as:

$$\begin{aligned}&\min_{\mathbf{A}} \|\mathbf{A}\|_F + c_S \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 - c_D \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2\\&\text{s. t. } \mathbf{A} \succeq 0 \qquad (8.5)\end{aligned}$$

where $\|\mathbf{A}\|_F$ stands for the Frobenius norm. If $\mathbf{A} = [a_{i,j}]_{m\times m}$, its Frobenius norm is define as:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i,j=1}^{m} a_{i,j}^2} \tag{8.6}$$

There are three items in (8.5). This item $\|\mathbf{A}\|_F$ serves as the regularization term for matrix $\mathbf{A}$, which prevents any elements within $\mathbf{A}$ from being too large. In particular, it prefers a sparse distance metric, in which many elements of $\mathbf{A}$ are zeros or close to zeros. A similar idea has been used in support vector machines [21], in which the L2 norm of hyper-plane weights is used for regularization. The second and third items in (8.5) represent the sum of squared distance between similar images and dissimilar images in log data. A discriminative distance metric $\mathbf{A}$ is learned such that similar images are close to each other in the space of image features and meanwhile dissimilar images are separated far away. Parameters c_S and c_D balance the tradeoff between the goal of minimizing distance among similar images and the goal of maximizing distance among dissimilar images. By adjusting these two parameters, we are also able to make a balanced trade-off between the robustness of the learned distance metric and the discriminative power of the metric. Note that, compared to (8.4), the new formulism in (8.5) moves the image pairs in the dissimilar set to the objective function. As a result, we relax the requirement on the image pairs in $\mathcal{D}_q$: instead of assuming that all image pairs in $\mathcal{D}_q$ *must* be dissimilar to each other, we only assume that they *could* be dissimilar to each other. Through this relaxation, we are able to improve the robustness of metric learning, particularly when there are a number of errors in the log data of users' relevance judgments.

Using the distance expression in (8.2), both the second and the third items of objective function in (8.5) can be expanded into the following forms:

$$\begin{aligned}
c_S \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 &= c_S \ \mathrm{tr}\left(\mathbf{A}\cdot \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\right) \\
&= c_S \sum_{i,j=1}^{m} a_{i,j} s_{i,j} \qquad (8.7)
\end{aligned}$$

and

$$\begin{aligned}
c_D \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 &= c_D \ \mathrm{tr}\left(\mathbf{A}\cdot \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\right) \\
&= c_D \sum_{i,j=1}^{m} a_{i,j} d_{i,j} \qquad (8.8)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{S} &= [s_{i,j}]_{m\times m} = \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \\
\mathbf{D} &= [d_{i,j}]_{m\times m} = \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T
\end{aligned}$$

As indicated in (8.7) and (8.8), both terms are linear in matrix $\mathbf{A}$.

Putting Eqn. (8.6), (8.7), (8.8) together, we have the final formulism for the regularized metric learning:

$$\begin{aligned}
&\min_{\mathbf{A}} \left(\sum_{i,j=1}^{m} a_{i,j}^2\right)^{1/2} + c_S \sum_{i,j=1}^{m} a_{i,j} s_{i,j} - c_D \sum_{i,j=1}^{m} a_{i,j} d_{i,j} \\
&\text{s. t. } \mathbf{A} \succeq 0 \qquad (8.9)
\end{aligned}$$

To convert the above problem into the standard form, we introduce a slack variable t that upper bounds the Frobenius norm of matrix $\mathbf{A}$, which leads to an equivalent form of (8.9), i.e.,

$$\begin{aligned}
&\min_{\mathbf{A},t} \ t + c_S \sum_{i,j=1}^{m} a_{i,j} s_{i,j} - c_D \sum_{i,j=1}^{m} a_{i,j} d_{i,j} \qquad (8.10) \\
&\text{s. t. } \left(\sum_{i,j=1}^{m} a_{i,j}^2\right)^{1/2} \leq t \\
&\qquad \mathbf{A} \succeq 0 \qquad (8.11)
\end{aligned}$$

In the above optimization problem, the objective function is linear in both t and $\mathbf{A}$. It has two constraints: the first constraint is called a second order cone constraint [140], and the second constraint is a positive semi-definite constraint. Both these two types of constraints are special forms of convex constraints. They have been well studied in the optimization theory [140], and there exist very efficient solutions that guarantee to solve this problem in a polynomial time (i.e., polynomial in m^2, the square of the number of low-level image features). Note that, in the formulism in 8.11, we allow matrix $\mathbf{A}$ to be in *any* form as long as it is symmetric and positive definitive. In this work, an interior-point optimization method implemented in the SeDuMi [129] optimization toolbox is used to solve the optimization problem in (8.11).

8.4 Experiment Methodology

8.4.1 Testbed

The collection of COREL image CDs contains a large number of real world images with semantic annotations. It has been widely used in previous CBIR research. In this work, two testbeds with images from 20 categories and 50 categories were created. Each category contains 100 images and is associated with specific semantic meaning such as antique, cat, dog and lizard, etc. Given a query image from the testbed, a retrieved image is considered to be relevant when it belongs to the same category of the query image. The average precision of top retrieved images is used to measure the quality of retrieved results. Despite that such a definition of relevance judgments may not accurately reflect the characteristics of relevance judgments by real-world users, it is able to avoid the subjectiveness in manual relevance judgments. Furthermore, it automates the process of evaluation and allows different approaches to be compared based on the same ground truth. In practice, this evaluation methodology has been adopted by many studies of image retrieval, such as [55, 56, 49, 136, 48, 50, 57].

8.4.2 Low-level Image Feature Representation

Low-level image feature representation is one of the key components for CBIR systems. Three types of visual features were used in this work, including color, edge and texture. The same set of image features have been used in the previous research on image retrieval [57].

- *Color* Three types of color moments were used: color mean, color variance and color skewness in three different color channels (i.e., H, S and V). Thus, totally nine different features were used to represent color information.

- *Edge* Edge features have been shown to be effective in CBIR since it provides information about shapes of different objects. The histogram for edge direction was first obtained by applying the Canny edge detector [65] to images. Then, the edge direction histogram was quantized into 18 bins of every 20 degrees, which resulted in totally 18 different edge features.

- *Texture* Texture is another type of popular features used in CBIR . In this work, we used texture features based on wavelet transformation. The Discrete Wavelet Transformation (DWT) was first applied to images with a Daubechies-4 wavelet filter [126]. 3-levels of wavelet decomposition were used to obtain ten subimages in different scales and orientations. One of the subimages is a subsampled average image of the original one and was discarded as it contains less useful information. The entropies of the other nine subimages were used to represent the texture information of images.

Therefore, altogether 36 features were used in this work to represent images.

8.4.3 Log Data of Users' Relevance Feedback

The log data of users' relevance feedback were collected from real world users of a CBIR system that is developed in the Chinese University of Hong Kong. 10 researchers participated in this experiment. In our experiment, for each log session, a sample query image was randomly generated. Given the query image, the CBIR system did retrieval by computing the Euclidean distance between the query image and images in database. The top 20 most similar images were returned to users. Users provided relevance judgement for each returned image by judging if it is relevant to the query image. Each user was asked to provide 10 or 15 log sessions on both the 20-category and the 50-category testbeds, respectively. All the feedback data from different log sessions were collected to build the users' log data.

An important issue for log data in real-world CBIR systems is that potentially users can make mistakes in judging the relevance of retrieved images. Thus, in reality there will be some amount of noise inside the log data of users' relevance feedback. Erroneous judgements can be caused by a variety of reasons, such as users' inconsistent and subjective judgments, and users' action mistakes. In order to evaluate the robustness of our algorithm, we collect log data with different amount of noises. The noise of log data is measured by its percentage of incorrect relevance judgments, i.e.,

$$P_{noise} = \frac{\text{Total number of wrong judgements}}{N_l \times N_{log}} \times 100\%$$

where N_l and N_{log} stand for the number of labeled examples acquired for each log session and the number of log sessions, respectively. To acquire log data with different amount of noise, we conduct experiments under two different setups. In the first setup, users' relevance

Table 8.1: The characteristics of log data collected from the real-world users

Datasets	Normal Log Data		Noisy Log Data	
	# Log Sessions	Noise (P_{noise})	# Log Sessions	Noise (P_{noise})
20-Category	100	7.8%	100	16.2%
50-Category	150	7.7%	150	17.1%

judgments are collected under normal behaviors of users, which leads to relatively small numbers of mistakes. In the second setup, users are requested to provide feedback within a very short period of time, which leads to relatively higher mistakes. The reason for such a study is twofold: first, through this study, we are able to estimate the amount of noise will be engaged in normal behaviors of real-world users; Second, the second noisy log data is valuable to evaluate the robustness of our algorithms. Table 8.1 shows the two sets of collected log data for both datasets with different amounts of noise from real-world users. In total, 100 log sessions are collected for the 20-Category and 150 log sessions for the 50-Category dataset. Based on these log data with different configurations, we will be able to evaluate the effectiveness, the robustness, and the scalability of our algorithm for metric learning.

We would like to emphasize that the log data used in this work is created by collecting judgments from **real world** users. This is different from the log data of simulated users in [50], which are generated by conducting automatic retrieval for sample query images and acquiring relevance judgments based on images' category information. The log data of simulated users in [50] did not consider the data noise problem, which makes it less representative for real world applications than the data used in this work.

8.5 Experimental Results

An extensive set of experiment results are presented in this section to illustrate the effectiveness, robustness, and scalability of our new regularized metric learning algorithm. Particularly, empirical studies were conducted to address the following three questions:

1. *How effective is our new algorithm in boosting the retrieval accuracy of a CBIR system by using the log data*? Experiments were conducted to compare the effectiveness of the distance metric learned by our new algorithm to the default Euclidean distance metric. We also compare the proposed metric learning algorithm to the algorithm in [153] for image retrieval, and to the manifold learning algorithm for CBIR that also uses log data [50].

2. *How does our new algorithm behave when the amount of users' relevance feedback is modest*? Experiments were conducted to study the effectiveness of our new algorithm by varying the size of the log data.

3. *How does our new algorithm behave when large amount of noise is present in the log data*? Experiments were conducted to study the effectiveness of our new algorithm with respect to different amount of noise.

8.5.1 Experiment I: Effectiveness

Four algorithms are compared in this section for their accuracy of image retrieval:

1. A baseline CBIR system that uses the Euclidean distance metric and does not utilize users' log data. We refer to this algorithm as "**Euclidean**".

2. A CBIR system that uses the semantic representation learned from the manifold learning algorithm in [50]. We refer to this algorithm as "**IML**".

3. A CBIR system that uses the distance metric learned by the algorithm in [153]. We refer to this algorithm as "**DML**".

4. A CBIR system that uses the distance metric learned by the proposed regularized metric learning algorithm. We refer to this algorithm as "**RDML**".

All the algorithms were implemented with MATLAB. Specifically, for the implementation of the manifold learning algorithm for image retrieval (i.e., IML), we followed the procedure described in [50]. All the parameters in the algorithm IML were carefully tuned to achieve good retrieval accuracy. For the algorithm based on metric learning in [153] (i.e., DML), we download the code from the web site of the author[1], and slightly modified the downloaded code to fit it in the CBIR task. Finally, the proposed algorithm based on regularized metric learning (i.e. RDML) was implemented within MATLAB using the SeDuMi optimization toolbox [129] to solve the optimization problem in (8.11). Parameter c_S in (8.11) was set to 0.15 and 0.1 for the 20-Catgory and the 50-Category testbeds, respectively. Another parameter c_D was set to be one third of Cs.

The experiment in this section was conducted for the log data with small noise, i.e., 7.8% noise for the 20-Category testbed, and 7.7% noise for the 50-Category testbed. All the users' log data were used in this experiment, i.e. 100 and 150 log sessions for 20-Category and 50-Category testbeds, respectively. Every image in the database was used as a query image. The results of mean average precision for the top-ranked images are reported in Tables 8.2 and 8.3. Several observations

[1] http://www-2.cs.cmu.edu/ẽpxing/publication.html

Table 8.2: Average precision (%) of top-ranked images on the 20-Category testbed over 2,000 queries. The relative improvement of algorithm IML, DML, and RDML over the baseline Euclidean is included in the parenthesis following the average accuracy.

Top Images	20	40	60	80	100
Euclidean	39.91	32.72	28.83	26.47	24.47
IML	42.66(6.9%)	34.32(4.9%)	30.00(4.1%)	26.47(0.3%)	23.80(-2.7%)
DML	41.45(3.9%)	34.89(6.6%)	31.21(8.2%)	28.63(8.5%)	26.44(8.0%)
RDML	44.55(11.6%)	37.39(14.3%)	33.11(14.8%)	30.13(14.1%)	27.82(13.7%)

Table 8.3: Average precision (%) of top-ranked images on the 50-Category testbed over 5,000 queries. The relative improvement of algorithm IML, DML, and RDML over the baseline Euclidean is included in the parenthesis following the average accuracy.

Top Images	20	40	60	80	100
Euclidean	36.39	28.96	24.96	22.21	20.18
IML	35.64(-2.1%)	29.16(0.7%)	24.75(-0.8%)	21.68(-2.4%)	19.32(-4.3%)
DML	33.52(-7.9%)	27.15(-6.3%)	23.77(-4.8%)	21.48(-3.3%)	19.74(-2.2%)
RDML	40.36(10.9%)	32.62(12.6%)	28.24(13.1%)	25.17(13.4%)	22.86(13.3%)

can be drawn from Tables 8.2 and 8.3:

- Compared to the baseline model, the manifold learning method (IML) gains a small improvement for the 20-Category testbed, but it fails to improve the retrieval accuracy of CBIR for the 50-Category testbed. One possible explanation is that the IML method does not explicitly explore the Min/Max principle when it is using the log data. In particular, it is only able to exploit the images that have been judged as relevant and is unable to utilize the images judged as irrelevant. Note that the empirical results for the IML algorithm reported in this work is not consistent with the results reported in [50], where the IML method achieves a significant improvement over the Euclidean distance metric. After consulting the authors for IML, we believe that the inconsistency could be attributed to different characteristics of log data used in these two studies. Not only was a much larger amount of users' log data used in [50] than in this work, but also their log data did not include any noise. To further confirm the correctness of this explanation, we followed the same procedure described in [50] and constructed similar log data of simulated users. We tested our implementation of the IML algorithm using the simulated log data and observed a similar amount of improvement as reported in [50]. Based on these results, we are confirmed that the IML algorithm works well when a large amount of log data is available. It may fail to improve the performance of CBIR when the size of log data is small.

- The distance metric learning (DML) algorithm does achieve certain amount of improvement over the baseline algorithm on the 20-Category testbed. But it performs consistently worse than the Euclidean distance on the 50-Category testbed. These results indicate that distance metric learned by the DML algorithm may not be robust and can suffer from the overfitting problem. This is because images from the 50-Category testbed are much more diverse than images from the 20-Category testbed. In contrast, the size of log data for the 50-Category testbed is only slightly larger than that for the 20-Category testbed. Thus, log data may not be sufficient for representing the diversity of the 50-Category testbed, which leads the DML algorithm to over-fit log data and therefore degrades the retrieval accuracy.

- Compared to the baseline method, the proposed algorithm for regularized distance metric learning (RDML) is able to consistently achieve more than 10% improvement in mean average precision for the top-ranked images. These results indicate that the RDML algorithm is more robust than the other two algorithms in boosting the retrieval accuracy of CBIR with log data. We attribute the success of the RDML algorithm to the combination of the discriminative training, which is based on the Min/Max principle, and the regularization procedure, which results in more robust distance metric.

To further illustrate the behavior of the RDML algorithm, we list the retrieval results of a sample query image in Figure 8.1. The first row of Figure 8.1 shows the top-5 returned images from the CBIR system using Euclidean distance metric, while the second row represents the results by the CBIR system using the distance metric learned by the RDML algorithm. The first image of each row is the sample query image. It can be seen that the CBIR system using Euclidean metric only acquired 2 relevant images (including the query image) out of top 5 returned images, while the CBIR system using the RDML algorithm did a better work by retrieving two more relevant images (the fourth one and the fifth image on the second row).

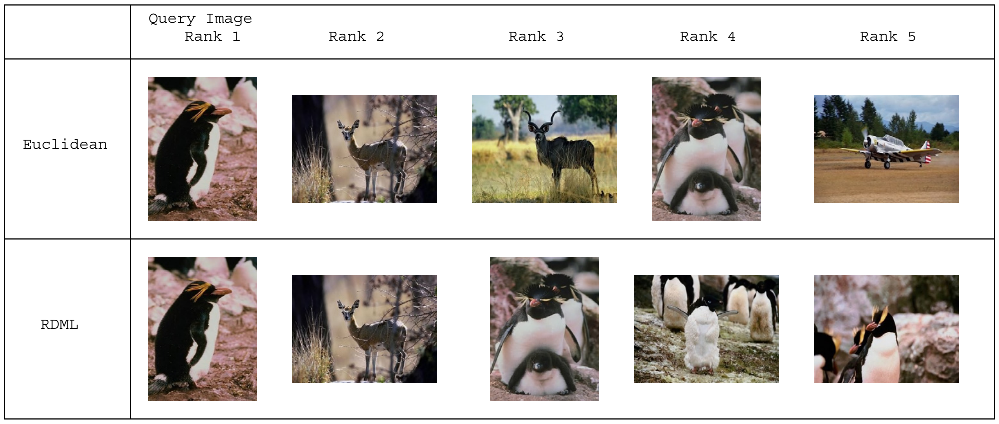


Figure 8.1: The retrieval results of top-5 returned images of a sample query image (the first one in the next two rows) for CBIR systems with either the Euclidean distance metric (first row) or the distance metric learned by RDML (second row).

Table 8.4: The training time cost (CPU seconds) of three algorithms on 20-Category (100 log sessions) and 50-Category (150 log sessions) testbeds.

Algorithm	**IML**	**DML**	**RDML**
20-Category	82.5	3,227	19.2
50-Category	2,864	12,341	20.5

8.5.2 Experiment II: Efficiency and Scalability

In addition to being more effective than the IML and the DML algorithm, the RDML algorithm can also be computed substantially more efficiently than the other two algorithms and is scalable to the size of log data. To manifest the efficiency and scalability of the proposed algorithm, we conducted a set of experiments to show the training time of these three algorithms. All the algorithms were run on a Windows XP operation system that is powered by a 2.0 GHz PC with 1GB physical memory. The training times of these three algorithms are shown in Table 8.4. As indicated in Table 8.4, the RDML algorithm can be trained much more efficiently than the other two algorithms for both testbeds. Particularly, two observations can be drawn from Table 8.4:

- The RDML algorithm is significantly more efficient than the DML algorithm. For both datasets, the training cost of the DML algorithm is at least two orders larger than that of the RDML algorithm. Note that both algorithms try to learn the distance metric $\mathbf{A}$ from the same log data and therefore have the same problem size. The RDML algorithm is more efficient than the DML algorithm because its related optimization problem can be solved efficiently by the SDP technique, while the DML algorithm has to solve a general convex programming problem that is usually much more time-consuming.

- The RDML algorithm is significantly more scalable to the size of log data than the IML algorithm. For the 20-Category testbed,

Table 8.5: Average precision (%) of top-ranked images on the 20-Category testbed for IML, DML, and RDML algorithm using small amounts of log data. The relative improvement over the baseline Euclidean is included in the parenthesis.

Top Images	20	40	60	80	100
Euclidean	39.91	32.72	28.83	26.47	24.47
IML (#Log 67)	39.01 (-2.3%)	31.49 (-3.8%)	27.64 (-4.1%)	24.75 (-6.5%)	22.43 (-8.3%)
DML (#Log 67)	41.03(2.8%)	34.73 (6.1%)	31.26 (8.4%)	28.67 (8.3%)	26.47 (8.2%)
RDML (#Log 67)	43.80(9.7%)	36.15(10.5%)	32.00(11.0%)	29.20(10.6%)	26.89(9.9%)
IML (#Log 33)	36.64(-8.2%)	29.72 (-9.2%)	25.99(-9.9%)	23.41(-11.6%)	21.53(-12.0%)
DML (#Log 33)	38.13 (-4.5%)	31.99(-2.2%)	28.69(-0.5%)	26.34 (-0.5%)	24.50 (-0.1%)
RDML (#Log 33)	42.56(6.6%)	35.12(7.3%)	31.01(7.5%)	28.17(6.7%)	26.11(6.7%)

both the IML algorithm and the RDML algorithm have similar training cost. However, for the 50-Category testbed, the training cost for the IML algorithm shoots up to about 3,000 Sec. Whereas the RDML algorithm is able to maintain its training cost almost unchanged between the 20-Category and the 50-Category. This is because the IML algorithm needs to solve a generalized eigenvalue decomposition problem [50], in which the problem size is not only dependent on the number of image features, but also dependent on the number of images in log data. Given the computational complexity of principle eigenvectors is on the order of n^3 where n is the number of variables, the IML algorithm cannot scale up to the size of log data. In contrast, the problem size for the RDML algorithm, only depends on the number of image features, thus is the same for both testbeds. As a result, regardless of the size of log data, the problem sizes of the RDML algorithm are the same, which leads to unchanged training cost.

8.5.3 Experiment III: Different Size of Log Data

In real world CBIR applications, it may be difficult to acquire large amount of users' log data. This issue is especially important in the early stage of system development. It is also important when the tar-

Table 8.6: Average precision (%) of top-ranked images on the 50-Category testbed for IML, DML, and RDML using small amounts of log data. The relative improvement over the baseline Euclidean is included in the parenthesis.

Top Images	20	40	60	80	100
Euclidean	36.39	28.96	24.96	22.21	20.18
IML (#Log 100)	34.25(-5.8%)	27.65(-4.5%)	23.34(-6.5%)	20.69(-6.8%)	18.49(-8.4%)
DML (#Log 100)	33.53(-7.9%)	26.84 (-7.3%)	23.28(-6.7%)	20.93(-5.8%)	19.21 (-4.8%)
RDML (#Log 100)	39.10(7.4%)	31.62(9.2%)	27.28(9.3%)	24.30(9.4%)	22.02(9.2%)
IML (#Log 50)	32.95(-9.5%)	26.87 (-7.2%)	22.92(-8.2%)	20.35 (-8.4%)	18.25 (-9.6%)
DML (#Log 50)	29.78(-18.2%)	23.26 (-19.7%)	19.86(-20.4%)	17.70(-20.3%)	16.13(-20.1%)
RDML (#Log 50)	38.96(7.1%)	31.44(8.6%)	27.08(8.5%)	24.09(8.5%)	21.76(7.9%)

get images are not popular and are only equipped with a few relevance judgments. In this case, the CBIR system has to provide retrieval service with limited amount of log data. A set of experiments was designed in this section to show the behavior of the RDML algorithm together with the IML and the DML algorithm in response to different size of log data. Different from the experiments presented in the previous sections, where all users' log data are used, in this section, all the algorithms were trained with only part of users' log data. In particular, it was trained with one-third and two-third users' log data for both testbeds. The empirical results are shown in Tables 8.5 and 8.6.

It can be seen from these two tables that the advantage of the RDML algorithm over the baseline algorithm using Euclidean distance metric decreases with less training data. However, even with very limited amount of training data, i.e. 33 log sessions for 20-Category and 50 log sessions for 50-Category, the RDML algorithm is still capable to gain notable improvement over the baseline model, which is about 7% for 20-Category and about 8% for 50-Category. Compared to the RDML algorithm, the IML algorithm and the DML algorithm suffer from substantially more degradation in the retrieval accuracy. In fact, for most cases when a small amount of log data is present, both the IML algorithm and the DML algorithm perform even worse than the

Table 8.7: Average precision (%) of top-ranked images on the 20-Category testbed for IML, DML, and RDML using noisy log data. The relative improvement over the baseline Euclidean is included in the parenthesis following the average accuracy.

Top Images	20	40	60	80	100
Euclidean	39.91	32.72	28.83	26.47	24.47
IML (Large Noise)	37.94(-4.9%)	30.14(-7.9%)	25.93(-10.1%)	23.56 (-11.0%)	21.97(-10.2%)
DML (Large Noise)	38.62(-3.2%)	32.32(-1.2%)	28.95(0.4%)	26.61 (0.8%)	24.62(0.6%)
RDML (Large Noise)	41.19(3.2%)	34.15(4.4%)	30.40(5.4%)	27.92(5.8%)	25.89(5.8%)

Table 8.8: Average precision (%) of top-ranked images on the 50-Category testbed for IML, DML, and RDML using noisy log data. The relative improvement over the baseline Euclidean is included in the parenthesis following the average accuracy.

Top Images	20	40	60	80	100
Euclidean	36.39	28.96	24.96	22.21	20.18
IML (Large Noise)	33.80(-7.1%)	27.30(-5.8%)	23.56(-5.0%)	20.65(-6.7%)	18.36 (-8.1%)
DML (Large Noise)	32.85(-9.7%)	26.95 (-7.0%)	23.55(-5.7%)	21.22(-4.5%)	19.49(-3.4%)
RDML (Large Noise)	37.45(2.9%)	29.97(3.5%)	25.84(3.5%)	22.99(3.5%)	20.87(3.4%)

straightforward Euclidean distance. In sum, this set of experiments demonstrates the robustness of the RDML algorithm in improving content-based image retrieval with the limited amount of users' log data, which can be important for real world CBIR systems.

8.5.4 Experiment IV: Noisy Log Data

Another practical problem with real-world CBIR applications is that the log data of user feedback are inevitable to contain certain amount of noise. The experiment results in previous sections have demonstrated that the RDML algorithm is able to boost the retrieval results of a CBIR system when log data have only a small amount of noise. It is interesting to investigate the behavior of the RDML algorithm when more noise is present in the log data of users' relevance feedback.

Experiments were conducted on both the 20-Category and the 50-Category testbeds using the log data that contain a large amount of noise. The details of users' log data with large noise have been described in Section 8.4.3. The experiment results for two testbeds using the RDML algorithm are shown in Tables 8.7 and 8.8, respectively. It can be seen from the experiment results that the noise in users' log data does have a significant impact on the retrieval accuracy, which is consistent with our expectation. However, even when the noisy log data that contain over 15% incorrect relevance judgments, the RDML algorithm still shows a consistent improvement over the baseline method using the Euclidean distance metric, although the improvement is small. In contrast, both the IML algorithm and the DML algorithm fail to improve the performance over the Euclidean distance when the log data is noisy. These results indicate the robustness of our new algorithm, which again is important for real-world CBIR applications.

8.6 Limitation and Future Work

Based on the promising results achieved from the above extensive empirical evaluations, we conclude that the regularized metric learning algorithm is effective for improving the performance of CBIR systems by integrating the log data of users' relevance feedback. Through the regularization mechanism, the learned distance metric is more robust. By formulating the learning problem into an SDP problem, it can be solved efficiently and is scalable to the size of log data. However, it is necessary to address the limitation and the challenging issues with the proposed algorithm as well as feasible directions for solving these problems in our future work.

First, we realize that the selection of parameter c_S and c_D in the proposed algorithm is important to its retrieval performance. Although our empirical approach for choosing c_S and c_D has resulted in good performance, we plan to investigate other principled approaches for effectively tuning these two parameters. One potential approach is to automatically determine these two parameters using the cross validation method. It divides the log data into 20%/80% partitions where 80% of the data is used for training and 20% for validation. The optimal values of c_S and c_D are found by maximizing the retrieval accuracy of the validation set.

Second, although our algorithm is robust to the noise present in the log data, the degradation in the retrieval accuracy caused by erroneous judgments is still quite significant. Hence, in the future, we plan to consider more sophisticated regularization approaches for metric learning, such as manifold regularization [15].

Third, in the proposed algorithm, a single distance metric is learned to describe the similarity between *any* two images. Given a heterogeneous collection that consists of multiple different types of images, a single distance metric may not be sufficient to account for diverse types of similarity functions. In the future, some interesting extensions can be naturally derived from our work. One possible way is to learn multiple query-dependent distance metrics with respect to different query types, which is similar to the idea of *query classification based retrieval* [72] in document information retrieval. Moreover, we may also learn multiple user-dependent distance metrics if users' preferences are available.

8.7 Summary

This chapter investigated a novel algorithm for distance metric learning, which boosts the retrieval accuracy of CBIR by taking advantage of the log data of users' relevance judgments. A regularization mechanism is used in the proposed algorithm to improve the robustness of solutions, when the log data is small and noisy. Meanwhile, it is formulated as an SDP problem, which can be solved efficiently and therefore is scalable to the size of log data.

Experiment results have shown that the proposed algorithm for regularized distance metric learning substantially improves the retrieval accuracy of the baseline CBIR system that uses the Euclidean distance metric. It is also more effective and more efficient than two alternative algorithms that also utilize the log data to enhance image retrieval. More empirical studies indicate that the new algorithm gains notable improvement even with limited amount of users' log data. Furthermore, the new algorithm is rather robust to work in the environment where the log data is noisy and contains a number of erroneous judgments. All these advantages make the new algorithm proposed in this chapter a good candidate for combining the log data and the low-level image features to improve the retrieval performance of CBIR systems.

□ **End of chapter.**

Chapter 9

Conclusion

9.1 Summary of Achievements

This thesis aimed to develop an unified scheme that is able to combine several statistical machine learning techniques for solving a variety of learning tasks in real-world applications effectively. To this purpose, we presented a novel general framework termed a unified learning paradigm (ULP), which integrates several learning methods in a synergistic way. Based on the idea of this unified learning framework, this thesis developed a novel scheme for learning Unified Kernel Machines (UKM) for classification tasks. In contrast to traditional classification approaches, UKM combines supervised kernel machine learning, unsupervised kernel design, semi-supervised kernel learning and active learning in a unified solution.

A key part of the UKM scheme is the development of an effective semi-supervised kernel learning method. To tackle this problem, we suggested a new algorithm called Spectral Kernel Learning (SKL), which is formulated into a QP problem that can be efficiently solved. Empirical evaluations on benchmark datasets have shown that our SKL algorithm is promising for learning effective kernels in both computational efficiency and classification performance.

Another important part of the ULP framework is the effective execution of active learning. Traditional active learning methods usually select a single example for labeling in each learning iteration, which usually require large re-training cost. To overcome this challenge, this thesis proposes a framework of Batch Mode Active Learning (BMAL) that is able to select a batch of the most informative examples for labeling. The BMAL task is formulated into a convex optimization problem and is solved by an efficient bound optimization algorithm. Extensive trials on text categorization have shown that our algorithm is more promising than traditional approaches.

Since the unified learning framework is a long-term research challenge, some important open issues were specifically investigated in this thesis. One of these is the issue of how to learn distance metrics from contextual information. We proposed new algorithms called Discriminative Component Analysis (DCA) and Kernel DCA to learn both linear and nonlinear distance metrics; these algorithms enjoy the merits of simplicity and effectiveness. Empirical evaluations on data clustering have demonstrated the effectiveness of our algorithms.

In addition to the methodology studies, we also investigated machine learning techniques in some real-world applications in data mining and multimedia information retrieval. The first application is the problem of mining users' historical query logs and click-through data in web search engines. We suggested marginalized kernel techniques to tackle similarity measure problems by kernel design methods. This application can be considered to be the kernel initialization step in our unified learning framework, in which we showed how to exploit domain knowledge to initialize an effective kernel.

Another application is Collaborative Multimedia Retrieval (CMR). We have investigated supervised kernel machine learning techniques and distance metric learning techniques to tackle two CMR learning tasks, i.e., online Log-based Relevance Feedback (LRF) and offline distance metric learning, in content-based image retrieval. Specifically, we proposed Soft Label Support Vector Machines (SLSVM) to solve the LRF problem and suggested the Regularized Distance Metric Learning (RDML) algorithm to develop a robust and scalable metric learning scheme for CMR. All of the above empirical demonstrations have shown that the unified learning framework and its extended algorithms are significant and important tools for addressing the challenging problems in real-world applications.

9.2 Future Work

Although a substantial number of promising achievements have been reported based on our novel framework, there are still numerous open issues that need to be further explored in future work. An important issue is to develop methodology for assessing the convergence problems of our unified learning framework. This challenge may be solved in a variety of ways, depending on the application. For example, given a classification task, it may be possible to evaluate error bounds or generalization performance of the unified kernel machines for convergence assessments.

The second challenge is the computational efficiency issue. Although we have developed efficient algorithms for each components in our unified learning framework, there are still some preprocessing steps those are computationally expensive. For example, in semi-supervised kernel learning, we need to conduct eigen-decompositions of an initial kernel matrix before the spectral kernel learning procedure; in batch mode active learning, we also need to do eigen-decompositions of the Fisher information matrix. For large-scale applications, the eigen-decomposition operation can be rather expensive. We can attack these challenges by exploring the sparsity property or studying more efficient eigen-decomposition algorithms.

Moreover, more study is needed of the scalability and generalization issue in the UKM framework. In our current solution, we assume the scheme is conducted in a transductive learning setting. Approaches for generalizing the scheme to unseen data and making it scalable for large-scale applications should be further studied. One solution is to search for an efficient method to learn the kernel matrix incrementally. Some approximation algorithms could probably be devised for solving the problem efficiently.

Last, but not least, we may extend our unified learning framework by exploring other existing machine learning techniques, such as reinforcement learning. Also we will apply our methodologies and algorithms to solve a variety of real-world applications in data mining and information retrieval, such as time-series regression, personalized web search, and biomedical data mining.

□ **End of chapter.**

Appendix A

Equation Derivation

A.1 Derivation of the Dual in SLSVM

The goal is to prove the derivation of the following optimization:

Definition 21 Soft Label Support Vector Machine (SLSVM):

$$\begin{aligned}
\min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_H \sum_{i=1}^{l} \xi_i + C_S \sum_{i=l+1}^{l+m} |s_i|\xi_i \qquad \text{(A.1)} \\
\text{subject to} \quad & y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - b) \geq 1 - \xi_i, \\
& \xi_i \geq 0 \ , i = 1, \ldots, l+m \ ,
\end{aligned}$$

where l and m are respectively the number of hard-labeled training data and the number of soft-label ones (with $|s_i| < 1$), C_H and C_S are weight parameters for hard-labeled and soft-labeled training data respectively. □

Let us introduce the positive Lagrange multipliers $\alpha_i, i = 1, 2, \ldots, l+m$, one for each of the inequality constraints in the above optimization problem, and μ_i for enforcing positivity of ξ_i. Then the Lagrangian functional can be formulated as follows:

$$\begin{aligned} L(\mathbf{w},\xi,b,\alpha,\mu) &= \frac{1}{2}\|\mathbf{w}\|^2 + C_H\sum_{i=1}^{l}\xi_i + C_S\sum_{i=l+1}^{l+m} y_i s_i \xi_i \\ &- \sum_{i=1}^{l}\alpha_i(y_i(\Phi(\mathbf{x}_i)\cdot\mathbf{w}-b)-1+\xi_i) \\ &- \sum_{i=l+1}^{l+m}\alpha_i(y_i(\Phi(\mathbf{x}_i)\cdot\mathbf{w}-b)-1+\xi_i) - \sum_{i=1}^{l+m}\mu_i\xi_i\ . \end{aligned} \tag{A.2}$$

By taking the partial derivative of L with respect to $\mathbf{w}$, $\xi_\mathbf{i}$, b and ρ, we can obtain the following equations respectively:

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l+m}\alpha_i y_i \Phi(\mathbf{x}_i) = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{l+m}\alpha_i y_i \Phi(\mathbf{x}_i)\ ; \\ &\forall i = 1,\ldots,l \\ &\frac{\partial L}{\partial \xi_i} = C_H - \alpha_i - \mu_i = 0 \Rightarrow 0 \le \alpha_i \le C_H\ , \\ &\forall i = l+1,\ldots,l+m \\ &\frac{\partial L}{\partial \xi_i} = y_i s_i C_S - \alpha_i - \mu_i = 0 \Rightarrow 0 \le \alpha_i \le y_i s_i C_S\ , \\ &\frac{\partial L}{\partial b} = \sum_{i=1}^{l+m}\alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{l+m}\alpha_i y_i = 0\ . \end{aligned}$$

By substituting the above equations into (A.2), one can derive the dual of the original optimization problem as follows:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{l+m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l+m}\alpha_i\alpha_j y_i y_j \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{l+m}\alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C_H\ , i = 1,2,\ldots,l\ , \\ & 0 \le \alpha_i \le y_i s_i C_S\ , i = l+1, l+2,\ldots,l+m\ . \end{aligned}$$

This finishes proving the derivation of the dual for the given optimization of OPT 2. ■

A.2 Derivation of Inequation in BMAL

Let $\mathcal{L}(\mathbf{q})$ be the objective function in (4.15). We then have

$$\begin{aligned}
\mathcal{L}(\mathbf{q}) &= \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^{n} q_i \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2} \\
&= \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^{n} q_i \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2} \times \frac{\sum_{i=1}^{n} q_i' \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2}{\sum_{i=1}^{n} q_i' \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2 \frac{q_i}{q_i'}}
\end{aligned}$$

Using the convexity property of reciprocal function, namely

$$\frac{1}{\sum_{i=1}^{n} p_i x} \leq \sum_{i=1}^{n} \frac{p_i}{x} \tag{A.3}$$

for $x \geq 0$ and p.d.f. $\{p_i\}_{i=1}^{n}$, we can arrive at the following deduction:

$$\begin{aligned}
&\frac{\sum_{i=1}^{n} q_i' \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2}{\sum_{i=1}^{n} q_i' \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2 \frac{q_i}{q_i'}} \\
&\leq \sum_{i=1}^{n} \frac{q_i' \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2}{\sum_{j=1}^{n} q_j' \pi_j (1-\pi_j)(\mathbf{x}_j^T \mathbf{v}_k)^2} \frac{1}{\frac{q_i}{q_i'}} \\
&= \sum_{i=1}^{n} \frac{(q_i')^2 \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2}{q_i \sum_{j=1}^{n} q_j' \pi_j (1-\pi_j)(\mathbf{x}_j^T \mathbf{v}_k)^2}
\end{aligned}$$

Substituting the above inequation back into (A.3), we can achieve the following inequality:

$$\begin{aligned}
\mathcal{L}(\mathbf{q}) &\leq \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^{n} q_i' \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2} \\
&\quad \times \left(\sum_{i=1}^{n} \frac{(q_i')^2 \pi_i (1-\pi_i)(\mathbf{x}_i^T \mathbf{v}_k)^2}{q_i \sum_{j=1}^{n} q_j' \pi_j (1-\pi_j)(\mathbf{x}_j^T \mathbf{v}_k)^2} \right) \\
&= \sum_{k=1}^{s} \frac{\lambda_k}{\left(\sum_{j=1}^{n} q_j' \pi_j (1-\pi_j)(\mathbf{x}_j^T \mathbf{v}_k)^2 \right)^2} \times \sum_{i=1}^{n} \frac{(q_i')^2 (\mathbf{x}_i^T \mathbf{v}_k)^2 \pi_i (1-\pi_i)}{q_i} \\
&= \sum_{i=1}^{n} \frac{(q_i'^2)}{q_i} \pi_i (1-\pi_i) \sum_{k=1}^{s} \frac{(\mathbf{x}_i \mathbf{v}_k)^2 \lambda_k}{(\sum_{j=1}^{n} q_j' \pi_j (1-\pi_j)(\mathbf{x}_j^T \mathbf{v}_k)^2)^2} \; .
\end{aligned}$$

This finishes the proof of the inequality mentioned above. ■

□ **End of chapter.**

Appendix B

List of Publications

1. **Steven C.H. Hoi**, Michael R. Lyu, Edward Y. Chang, "Learning the Unified Kernel Machines for Classification," In The Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2006), Philadelphia, USA, August 20 - 23, 2006. (acceptance rate=11%)

2. **Steven C.H. Hoi**, R. Jin, J. Zhu and M.R. Lyu, "Batch Mode Active Learning and Its application to Medical Image Classification", In Proceedings The 23rd International Conference on Machine Learning (ICML2006), Pittsburgh, Penn, US, June 25-29, 2006. (acceptance rate=18%)

3. **Steven C.H. Hoi**, Rong Jin and Michael R. Lyu, "Large-Scale Text Categorization by Batch Mode Active Learning," In Proceedings 15th International World Wide Web conference (WWW2006), Edinburgh, England, UK, 2006. (acceptance rate=11%)

4. Qiankun Zhao, **Steven C.H. Hoi**, T.-Y. Liu, S. S. Bhowmick, M.R. Lyu and W.-Y. Ma, "Time-Dependent Semantic Similarity Measure of Queries Using Historical Click-Through Data," In Proceedings 15th International World Wide Web conference (WWW2006), Edinburgh, England, UK, 2006. (acceptance rate=11%)

5. **Steven C.H. Hoi**, Wei Liu, Michael R. Lyu, Wei-Ying Ma, "Learning Distance Functions with Contextual Constraints for Image Retrieval," In Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2006), New York, June, 2006. (acceptance rate=20%)

6. Jianke Zhu, **Steven C.H. Hoi**, and Michael R. Lyu, "Real-Time Non-Rigid Shape Recovery via Active Appearance Models for Augmented Reality," In the 9th European Conference on Computer Vision (ECCV 2006), Graz, Austria May 7-13, 2006. (acceptance rate=21%)

7. **Steven C.H. Hoi**, Michael R. Lyu and Rong Jin, "A Unified Log-based Relevance Feedback Scheme for Image Retrieval," In IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 18, no. 4, pp.509–524, 2006. (Journal Publication)

8. Luo Si, Rong Jin and **Steven C. H. Hoi** and Michael R. Lyu, "Collaborative Image Retrieval via Regularized Metric Learning," ACM Multimedia Systems Journal (MMSJ), Special issue on Machine Learning Approaches to Multimedia Information Retrieval, 2006. (Journal Publication)

9. **Steven C.H. Hoi**, Jianke Zhu and Michael R. Lyu, "CUHK at ImageCLEF 2005: Cross-Language and Cross-Media Image Retrieval," In Proceedings of Cross Language Evaluation Forum (CLEF) campaign, Lecture Notes of Computer Science (LNCS), 2006

10. Jianke Zhu, **Steven C.H. Hoi**, Edward Yau and Michael R. Lyu, "Automatic 3D Face Modeling Using 2D Active Appearance Models," Pacific Graphics 2005 (PG 2005), Macau, China

11. **Steven C.H. Hoi**, Michael R. Lyu and Rong Jin, "Integrating

User Feedback Log into Relevance Feedback by Coupled SVM for Content-based Image Retrieval," (Invited paper/talk) In 1st IEEE International EMMA Workshop in conjunction with 21st ICDE Conference, Tokyo, Japan, 2005

12. Edward Y. Chang, **Steven C.H. Hoi**, Xinjing Wang, Wei-Ying Ma and Michael R. Lyu, "A Unified Machine Learning Paradigm for Large-Scale Personalized Information Management," (Invited paper/talk) The 5th Emerging Information Technology Conference (EIT2005), NTU Taipei, August 2005.

13. **Steven C.H. Hoi** and Michael R. Lyu, "A Semi-Supervised Active Learning Framework for Image Retrieval," In Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2005), San Diego, CA, USA June 20-25, 2005. (acceptance rate=20%)

14. **Chu-Hong Hoi** and Michael R. Lyu, "A Novel Log-based Relevance Feedback Technique in Content-based Image Retrieval," In Proceedings of the 12th ACM International Conference on Multimedia (MM2004), New York, USA, 10-16 October, 2004, pp. 24-31. (acceptance rate=16%)

15. **Chu-Hong Hoi** and Michael R. Lyu, "Web Image Learning for Searching Semantic Concepts in Image Databases," Alternative Track and Poster Proceedings of the 13th International World Wide Web Conference (WWW2004), New York, USA, 17-22 May, 2004

16. **Chu-Hong Hoi** and Michael R. Lyu, "Group-based Relevance Feedback with Support Vector Machine Ensembles," In Proceedings of the 17th International Conference on Pattern Recognition (ICPR2004), Cambridge, UK, 23-26 August, vol. 3, pp. 874-877, 2004

17. **Chu-Hong Hoi**, Chi-Hang Chan, Kaizhu Huang, Michael R Lyu and Irwin King, "Biased Support Vector Machine for Relevance Feedback in Image Retrieval," Proceedings of International Joint Conference on Neural Networks (IJCNN2004), Budapest, Hungary, 25-29 July, pp.3189-3194, 2004. (oral, acceptance rate=30%)

18. **Chu-Hong Hoi** and Michael R. Lyu, "Robust Face Recognition Using Minimax Probability Machine," In Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME2004), Taiwan, 27-30 June, 2004

19. **Chu-Hong Hoi**, Wei Wang and Michael R. Lyu, "A Novel Scheme for Video Similarity Detection," In Proceedings of International Conference on Image and Video Retrieval (CIVR2003), USA, pp. 373-382, LNCS, vol. 2728, Springer, 2003

20. **Steven C.H. Hoi**, Rong Jin and Michael R. Lyu, "Learning Nonparametric Kernels," CUHK Technical Report (submitted to a conference), 2006

□ **End of chapter.**

Bibliography

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *ACM Symposium on Theory of Computing (STOC)*, 2001.

[2] E. Z. B. Anderson. *LAPACK User's Guide (3rd Edition)*. Philadelphia, PA, SIAM, 1999.

[3] P. Anick. Using terminological feedback for web search refinement: a log-based study. In *Proceedings of The 26th Annual International ACM SIGIR Conference (SIGIR 2003)*, pages 88–95, 2003.

[4] C. Apte, F. Damerau, and S. Weiss. Automated learning of decision rulesfor text categorization. *ACM Trans. on Information Systems*, 12(3):233–251, 1994.

[5] T. V. Ashwin, J. Navendu, and S. Ghosal. Improving image retrieval performance with negative relevance feedback. In *Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing (ICASSP'01)*, Utah, USA, 2001.

[6] R. Attar and A. S. Fraenkel. Local feedback in full-text retrieval systems. *Journal of ACM*, 24(3):397–417, 1977.

[7] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research (JMLR)*, 3:1–48, 2003.

[8] R. A. Baeza-Yates, R. Baeza-Yates, and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[9] A. Bar-hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005.

[10] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 407–416, 2000.

[11] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder. Hourly analysis of a very large topically categorized web query log. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 321–328, 2004.

[12] M. Belkin and I. M. andd P. Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, 2004.

[13] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, 2002.

[14] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 2004.

[15] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from examples. In *University of Chicago CS Technical Report TR-2004-06*, 2004.

[16] S. Berretti, A. D. Bimbo, and P. Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on Multimedia*, (4):225–239, 2000.

[17] D. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the 26th Intl. ACM SIGIR Conf. (SIGIR'03)*, pages 127–134, 2003.

[18] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[19] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003.

[20] K. Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of international conference on Machine learning (ICML'03)*, Washington DC, USA, 2003. ACM Press.

[21] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[22] C. Campbell, N. Cristianini, and A. J. Smola. Query learning with large margin classifiers. In *17th International Conference on Machine Learning (ICML)*, pages 111–118, San Francisco, CA, USA, 2000.

[23] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[24] E. Chang, S. C. Hoi, X. Wang, W.-Y. Ma, and M. Lyu. A unified machine learning framework for large-scale personalized information management. In *The 5th Emerging Information Technology Conference*, NTU Taipei, 2005.

[25] E. Chang and M. Lyu. Unified learning paradigm for web-scale mining. In *Snowbird Machine Learning Workshop*, 2006.

[26] O. Chapelle, A. Zien, and B. Scholkopf. *Semi-supervised learning.* MIT Press, 2006.

[27] S. Chien and N. Immorlica. Semantic similarity between search engine queries using temporal correlation. In *Proceedings of the 14th international conference on World Wide Web*, pages 2–11, 2005.

[28] W. W. Cohen. Text categorization and relational learning. In *12th International Conference on Machine Learning (ICML)*, pages 124–132, 1995.

[29] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 705–712. The MIT Press, 1995.

[30] I. J. Cox, M. Miller, T. Minka, and P. Yianilos. An optimized interaction strategy for bayesian relevance feedback. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pages 553–558, Santa Barbara, CA, USA, 1998.

[31] N. Cristianini, J. Shawe-Taylor, and A. Elisseeff. On kernel-target alignment. *JMLR*, 2002.

[32] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web*, pages 325–332, 2002.

[33] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma;. Query expansion by mining user logs. *IEEE Transactions on Knowledge and Data Engineering*, (4):829–839, 2003.

[34] S. Deerwester, S.Dumais, G.Furnas, T.Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[35] P. Drineas and M. Mahoney. On the nystrom method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[36] P. Duygulu, K. Barnard, J. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proc. the 7th European Conf. on Computer Vision*, pages 97–112, 2002.

[37] K.-S. G. E. Y. Chang, S. Tong and C.-W. Chang. Support vector machine concept-dependent active learning for image retrieval. *To appear in IEEE Transactions on Multimedia*, 2005.

[38] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

[39] S. Fine, R. Gilad-Bachrach, and E. Shamir. Query by committee, linear separation and random walks. *Theor. Comput. Sci.*, 284(1):25–51, 2002.

[40] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Mach. Learn.*, 28(2-3):133–168, 1997.

[41] P. E. Gill, W. Murray, and M. Wright. *Practical Optimization.* Academic Press, London, 1981.

[42] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Proc. NIPS*, pages 513–520, 2004.

[43] Y. Gong, H. J. Zhang, H. C. Chuan, and M. Sakauchi. An image database system with content capturing and fast image indexing abilities. In *IEEE International Conference on Multimedia Computing and Systems*, pages 121–130, May 1994.

[44] T. Graepel and R. Herbrich. The kernel gibbs sampler. In *Advances in Neural Information Processing Systems 13*, pages 514–520, 2000.

[45] H. P. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, and V. Vapnik. Parallel support vector machines: The cascade svm. In *Advances in Neural Information Processing Systems*, 2005.

[46] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

[47] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification and regression. In *Proc. NIPS*, pages 409–415, 1996.

[48] J. He, M. Li, H. J. Zhang, H. Tong, and C. Zhang. Manifold ranking based image retrieval. In *Proceedings of ACM International Conference on Multimedia*, pages 9–16, 2004.

[49] X. He, O. King, W.-Y. Ma, M. Li, and H. J. Zhang. Learning a semantic space from user's relevance feedback for image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1):39–48, Jan. 2003.

[50] X. He, W.-Y. Ma, and H.-J. Zhang. Learning an image manifold for retrieval. In *Proceedings of the 12th ACM International Conference on Multimedia (MM2004)*, pages 17–23, New York, US, 2004.

[51] T. Heath, E. Motta, and M. Dzbor. Uses of contextual information to support online tasks. In *14th international conference on World Wide Web (WWW2005)*, pages 1102–1103, 2005.

[52] D. Heesch, A. Yavlinsky, and S. Rüuger. Performance comparison between different similarity models for cbir with relevance

feedback. In *Proceedings of International Conference on Image and Video Retrieval (CIVR'03)*, LNCS 2728, pages 456–466. Springer-Verlag, 2003.

[53] T. Hertz, N. Shental, A. Bar-Hillel, and D. Weinshall. Enhancing image and video retrieval: Learning via equivalence constraints. In *Proc. IEEE CVPR*, pages 668–674, 2003.

[54] C. H. Hoi and M. R. Lyu. Biased support vector machine for relevance feedback in image retrieval. In *Proceedings International Joint Conference on Neural Networks (IJCNN'04)*, pages 3189–3194, Budapest, Hungary, 2004.

[55] C. H. Hoi and M. R. Lyu. Group-based relevance feeedback with support vector machine ensembles. In *Proceedings 17th International Conference on Pattern Recognition (ICPR'04)*, pages 874–877, Cambridge, UK, 2004.

[56] C. H. Hoi and M. R. Lyu. Web image learning for searching semantic concepts in image databases. In *Poster Proc. 13th International World Wide Web Conference (WWW'2004)*, pages 406–407, New York, USA, 2004.

[57] C. H. Hoi and M. R. Lyu. A novel log-based relevance feedback technique in content-based image retrieval. In *Proceedings ACM Multimedia Conference (MM 2004)*, pages 10–16, New York, October 2004.

[58] C. H. Hoi, M. R. Lyu, and R. Jin. Integrating user feedback log into relevance feedback by coupled svm for content-based image retrieval. In *Proceedings of the 1st IEEE International Workshop on Managing Data for Emerging Multimedia Applications (EMMA 2005)*, 2005.

[59] S. C. Hoi, M. R. Lyu, and R. Jin. A unified log-based relevance feedback scheme for image retrieval. *IEEE Trans. on Knowledge and Data Engineering*, 18(4):509–524, 2006.

[60] P. Hong, Q. Tian, and T. S. Huang. Incorporate support vector machines to content-based image retrieval with relevant feedback. In *Proc. IEEE International Conference on Image Processing (ICIP'00)*, Vancouver, BC, Canada, 2000.

[61] T. S. Huang and X. S. Zhou. Image retrieval by relevance feedback: from heuristic weight adjustment to optimal learning methods. In *Proceedings of IEEE International Conference on Image Processing (ICIP'01)*, Thessaloniki, Greece, Oct. 2001.

[62] Y. Ishikawa, R. Subramanya, and C. Faloutsos. MindReader: Querying databases through multiple examples. In *Proc. 24th Int. Conf. Very Large Data Bases (VLDB'98)*, pages 218–227, 1998.

[63] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Proc. NIPS*, 1998.

[64] A. K. Jain and M. N. Murty. Data clustering: A review. *ACM Computing Surveys*, 32(3):264–323, 1999.

[65] A. K. Jain and A. Vailaya. Shape-based retrieval: a case study with trademark image database. *Pattern Recognition*, (9):1369–1390, 1998.

[66] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th Intl. ACM SIGIR Conference (SIGIR'03)*, pages 119–126, 2003.

[67] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proc. 10th European*

Conference on Machine Learning (ECML), number 1398, pages 137–142, 1998.

[68] T. Joachims. Making large-scale svm learning practical. In *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1999.

[69] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. 16th International Conference on Machine Learning (ICML)*, pages 200–209, San Francisco, CA, USA, 1999.

[70] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

[71] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, 2005.

[72] I.-H. Kang and G. Kim. Query type classification for web document retrieval. In *Proc. of the 26th ACM SIGIR Conference*, pages 64–71, 2003.

[73] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *20th International conference on Machine Learning (ICML)*, pages 321–328, 2003.

[74] P. Komarek and A. Moore. Fast robust logistic regression for large sparse datasets with binary outputs. In *Artificial Intelligence and Statistics (AISTAT)*, 2003.

[75] P. Komarek and A. Moore. Making logistic regression a core data mining tool: A practical investigation of accuracy, speed, and simplicity. In *Technical Report TR-05-27 at the Robotics Institute, Carnegie Mellon University*, May 2005.

[76] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7, pages 231–238. The MIT Press, 1995.

[77] J. Kwok and I. Tsang. Learning with idealized kernels. In *Proceedings of the 20th International Conference on Machine Learning*, pages 400–407, 2003.

[78] J. Laaksonen, M. Koskela, and E. Oja. Picsom: Self-organizing maps for content-based image retrieval. In *Proc. International Joint Conference on Neural Networks (IJCNN'99)*, Washington, DC, USA, 1999.

[79] M. Lan, C. L. Tan, H.-B. Low, and S. Y. Sung. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In *Posters Proc. 14th International World Wide Web Conference*, pages 1032–1033, 2005.

[80] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. Jordan. Learning the kernel matrix with semi-definite programming. *JMLR*, 5:27–72, 2004.

[81] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Advances in Neural Information Processing Systems (NIPS'03)*, 2003.

[82] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse gaussian process methods: The informative vector machine. In *Advances*

in Neural Information Processing Systems (NIPS), pages 625–632, 2003.

[83] G. Lebanon. Learning riemannian metrics. In *Proceedings of the 19th Conference on Uncertainty in Articial Intelligence*, 2003.

[84] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proc.17th ACM International SIGIR Conference*, pages 3–12, 1994.

[85] R. Liere and P. Tadepalli. Active learning with committees for text categorization. In *Proceedings 14th Conference of the American Association for Artificial Intelligence (AAAI)*, pages 591–596, MIT Press, 1997.

[86] Q. Liu, H. Lu, and S. Ma. Improving kernel fisher discriminant analysis for face recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(1):42–49, 2004.

[87] T.-Y. Liu, Y. Yang, H. Wan, Q. Zhou, B. Gao, H. Zeng, Z. Chen, , and W.-Y. Ma. An experimental study on large-scale web categorization. In *Posters Proceedings of the 14th International World Wide Web Conference*, pages 1106–1107, 2005.

[88] Y. Lu, C. Hu, X. Zhu, H. J. Zhang, and Q. Yang. A unified framework for semantics and feature based relevance feedback in image retrieval systems. In *Proceedings of the eighth ACM International Conference on Multimedia*, pages 31–37, New York, NY, USA, 2000. ACM Press.

[89] S. MacArthur, C. Brodley, and C. Shyu. Relevance feedback decision trees in content-based image retrieval. In *Proc. IEEE Workshop on Content-based Access of lmage and Video Libraries*, pages 68–72, 2000.

[90] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[91] B. Manjunath, P. Wu, S. Newsam, and H. Shin. A texture descriptor for browsing and similarity retrieval. *Signal Processing Image Communication*, 2001.

[92] B. Masand, G. Lino, and D. Waltz. Classifying news stories using memory based reasoning. In *15th ACM SIGIR Conference*, pages 59–65, 1992.

[93] A. K. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *Proc.15th International Conference on Machine Learning*, pages 350–358. San Francisco, CA, 1998.

[94] G. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley, 1992.

[95] P. Melville and R. J. Mooney. Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning (ICML'04)*, page 74, New York, NY, USA, 2004. ACM Press.

[96] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher discriminant analysis with kernels. In *Proc. IEEE NN for Signal Processing Workshop*, pages 41–48, 1999.

[97] H. Muller, T. Pun, and D. Squire. Learning from user behavior in image retrieval: Application of market basket analysis. *Int. J. Comput. Vision*, 56(1-2):65–77, 2004.

[98] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *In Advances in Neural Information Processing Systems 14*, 2001.

[99] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14*, pages 841–848, 2001.

[100] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39:103–134, 2000.

[101] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Machines*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[102] K. Porkaew, K. Chakrabarti, and S. Mehrotra. Query refinement for multimedia retrieval and its evaluation techniques in mars. In *Proceedings of ACM International Conference on Multimedia*, Orlando, Florida, USA, 1999.

[103] K. Porkaew, K. Chakrabarti, and S. Mehrotra. Query refinement for multimedia retrieval and its evaluation techniques in mars. In *Proceedings of ACM International Conference on Multimedia*, Orlando, Florida, USA, 1999.

[104] K. Porkaew, M. Ortega, and S. Mehrotra. Query reformulation for content based multimedia retrieval in MARS. In *Proceedings of ICMCS*, volume 2, pages 747–751, 1999.

[105] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248, 2005.

[106] J. Rocchio. Relevance feedback in information retrieval. *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323, 1971.

[107] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised selftraining of object detection models. In *Seventh IEEE Workshop on Applications of Computer Vision.*, 2005.

[108] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *18th International Conference on Machine Learning (ICML)*, pages 441–448, 2001.

[109] Y. Rui and T. S. Huang. A novel relevance feedback technique in image retrieval. In *Proceedings of ACM International Conference on Multimedia*, pages 67–70, Orlando, Florida, USA, 1999.

[110] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, Sept. 1998.

[111] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, Sept. 1998.

[112] M. E. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1):87–118, 2002.

[113] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[114] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 44(4):288–287, 1990.

[115] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, 1983.

[116] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proc. 17th International Conference on Machine Learning*, pages 839–846, 2000.

[117] B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[118] M. Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, 2001.

[119] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Computational Learning Theory*, pages 287–294, 1992.

[120] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[121] X. Shen, S. Dumais, and E. Horvitz. Analysis of topic dynamics in web search. In *14th international conference on World Wide Web (WWW2005)*, pages 1102–1103, 2005.

[122] L. K. Shih and D. R. Karger. Using urls and table layout for web classification tasks. In *Proc. International World Wide Web Conference*, pages 193–202, 2004.

[123] S. D. Silvey. *Statistical Inference.* 1974.

[124] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early

years. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[125] B. Smith, P. Bjrstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, 1996.

[126] J. Smith and S.-F. Chang. Automated image retrieval using color and texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Nov. 1996.

[127] A. Smola and R. Kondor. Kernels and regularization on graphs. In *Intl. Conf. on Learning Theory*, 2003.

[128] J. Sturm. Using sedumi: a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[129] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Special issue on Interior Point Methods (CD supplement with software).

[130] J. Sun, H.-J. Zeng, H. Liu, Y.-C. Lu, and Z. Chen. Cubesvd: A novel approach to personalized web search. In *Proceedings of international conference on World Wide Web (WWW2005)*, 2005.

[131] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, 2001.

[132] D. Tao and X. Tang. Nonparametric discriminant analysis in relevance feedback for content-based image retrieval. In *Proceedings IEEE International Conference on Pattern Recognition (ICPR)*, 2004.

[133] D. Tao and X. Tang. Random sampling based svm for relevance feedback image retrieval. In *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[134] K. Tieu and P. Viola. Boosting image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, South Carolina, USA, 2000.

[135] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proc. Annual Allerton Conf. on Communication, Control and Computing*, pages 368–377, 1999.

[136] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM Press.

[137] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Proc. 17th International Conference on Machine Learning (ICML)*, pages 999–1006, Stanford, US, 2000.

[138] K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(90001):268–275, 2002.

[139] K. Tzeras and S. Hartmann. Automatic indexing based on Bayesian inference networks. In *Proc. 16th ACM Int. SIGIR Conference*, pages 22–34, 1993.

[140] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[141] V. N. Vapnik. *Statistical Learning Theory.* John Wiley, 1998.

[142] N. Vasconcelos and A. Lippman. Learning from user feedback in image retrieval systems. In *Advances in Neural Information Processing Systems*, 1999.

[143] N. Vasconcelos and A. Lippman. Bayesian relevance feedback for content-based image retrieval. In *Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries (CVPR'00)*, South Carolina, USA, 2000.

[144] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 131–142, 2004.

[145] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. 18th ICML*, pages 577–584, 2001.

[146] L. Wang, C. Wang, X. Xie, J. Forman, Y. Lu, W.-Y. Ma, and Y. Li. Detecting dominant locations from search queries. In *20th International conference on Machine Learning (ICML)*, pages 321–328, 2003.

[147] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Clustering user queries of a search engine. In *Proceedings of the 10th international conference on World Wide Web*, pages 162–168, 2001.

[148] C. K. I. Williams. On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, 46(1–3):11–19, 2002.

[149] G. Wu, E. Y. Chang, Y.-K. Chen, and C. Hughes. Incremental approximate matrix factorization for speeding up support vector machines. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.

[150] G. Wu, N. Panda, and E. Y. Chang. Formulating context-dependent similarity functions. In *ACM International Conference on Multimedia (MM)*, pages 725–734, 2005.

[151] G. Wu, Z. Zhang, and E. Y. Chang. Kronecker factorization for speeding up kernel machines. In *SIAM International Conference on Data Mining (SDM)*, Newport Beach, 2005.

[152] Y. Wu, Q. Tian, and T. S. Huang. Discriminant-em algorithm with application to image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, South Carolina, USA, 2000.

[153] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512, Cambridge, MA, 2003. MIT Press.

[154] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of The 19th Annual International ACM SIGIR Conference (SIGIR'96)*, pages 4–11, 1996.

[155] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.*, 18(1):79–112, 2000.

[156] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan. Optimizing web search using web click-through data. In *Proceedings of the thirteenth ACM conference on Information and knowledge management*, pages 118–126, 2004.

[157] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.

[158] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings 14th International Conference on Machine Learning (ICML)*, pages 412–420, Nashville, US, 1997.

[159] H. Yu and J. Yang. A direct lda algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34:2067–2070, 2001.

[160] J. Zhang, R. Jin, Y. Yang, and A. Hauptmann. Modified logistic regression: An approximation to svm and its applications in large-scale text categorization. In *Proc. 20th International Conference on Machine Learning (ICML)*, Washington, DC, USA, 2003.

[161] K. Zhang and J. T. Kwok. Block-quantized kernel matrix for fast spectral embedding. In *Proceedings of International Conference on Machine learning (ICML'06)*, 2006.

[162] L. Zhang, F. Lin, and B. Zhang. Support vector machine learning for image retrieval. In *Proceedings International Conference on Image Processing (ICIP2001)*, volume 2, pages 721–724, 2001.

[163] T. Zhang and R. K. Ando. Analysis of spectral kernel design based semi-supervised learning. In *NIPS*, 2005.

[164] T. Zhang and F. J. Oles. A probability analysis on the value of unlabeled data for classification problems. In *17th International Conference on Machine Learning (ICML)*, 2000.

[165] Q. Zhao, S. C. H. Hoi, T.-Y. Liu, S. S. Bhowmick, M. R. Lyu, and W.-Y. Ma. Time-dependent semantic similarity measure of queries using historical click-through data. In *Proceedings of the 15th International World Wide Web conference (WWW2006)*, Edinburgh, England, UK, May 23–26 2006.

[166] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schlkopf. Learning with local and global consistency. In *NIPS'16*, 2005.

[167] X.-D. Zhou, L. Zhang, L. Liu, Q. Zhang, and B.-L. Shi. A relevance feedback method in image retrieval by analyzing feedback log file. In *Proc. International Conference on Machine Learning and Cybernetics*, volume 3, pages 1641–1646, Beijing, 2002.

[168] X. S. Zhou, A. Garg, and T. S. Huang. A discussion of nonlinear variants of biased discriminants for interactive image retrieval. In *CIVR 2004 (LNCS 3115)*, pages 353–364, 2004.

[169] X. S. Zhou and T. S. Huang. Small sample learning during multimedia retrieval using biasmap. In *Proc. IEEE CVPR*, 2001.

[170] J. Zhu. Semi-supervised learning literature survey. Technical report, Carnegie Mellon University, 2005.

[171] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. In *NIPS 14*, pages 1081–1088, 2001.

[172] J. Zhu and J. Lafferty. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proc. 22nd ICML*, 2005.

[173] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings International Conference on Machine Learning (ICML2003)*, 2003.

[174] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. ICML'2003*, 2003.

[175] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *NIPS2005*, 2005.